国外计算机科学教材系列

支持向量机导论

An Introduction to Support Vector Machines
and Other Kernel-based Learning Methods

[英] Nello Cristianini
John Shawe-Taylor 著

李国正 王 猛 曾华军 译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

支持向量机（SVM）是在统计学习理论的基础上发展起来的新一代学习算法，该算法在文本分类、手写识别、图像分类、生物信息学等领域中获得了较好的应用。本书是第一本支持向量机方面的导论型读物。它从机器学习算法的基本问题开始，循序渐进地介绍相关的背景知识，包括线性分类器、核函数特征空间、推广性理论和优化理论，从而很自然地引出了支持向量机的算法。书的末尾还详细讨论了一系列支持向量机的重要应用以及实现的技巧。该书提供的大量相关文献以及网站链接为进一步学习提供了有效线索，有助于读者及时跟踪该领域的最新信息。本书可作为计算机、自动化、机电工程、应用数学等专业的研究生教材，也可作为神经网络、机器学习、数据挖掘等课程的参考教材，同时还是相关领域的教师和研究人员的参考书。

版权贸易合同登记号　图字：01-2002-4550

图书在版编目（CIP）数据

支持向量机导论 /（英）克里斯特安尼（Cristianini, N.）等著；李国正，王猛，曾华军译.
-北京：电子工业出版社，2004.3
（国外计算机科学教材系列）
书名原文：An Introduction to Support Vector Machines and Other Kernel-based Learning Methods
ISBN 7-5053-9336-7

Ⅰ.支... Ⅱ.①克... ②李... ③王... ④曾... Ⅲ.电子计算机-算法理论-教材 Ⅳ.TP301.6

中国版本图书馆CIP数据核字（2004）第009806号

责任编辑：史　平
印　　刷：北京兴华印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787 × 980　1/16　　印张：11　字数：215千字
印　　次：2004年3月第1次印刷
定　　价：25.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

译 者 序

支持向量机由 Vapnik 及其合作者发明，在 1992 年计算学习理论的会议上介绍进入机器学习领域，之后受到了广泛的关注。在 20 世纪 90 年代中后期得到了全面深入的发展，现已成为机器学习和数据挖掘领域的标准工具。

支持向量机是机器学习领域若干标准技术的集大成者。它集成了最大间隔超平面、Mercer 核、凸二次规划、稀疏解和松弛变量等多项技术。在若干挑战性的应用中，获得了目前为止最好的性能。在美国科学杂志上，支持向量机以及核学习方法被认为是“机器学习领域非常流行的方法和成功的例子，并是一个十分令人瞩目的发展方向”。

《支持向量机导论》是一本综合性介绍支持向量机各项标准技术的著作，书中从学习方法到超平面、核函数、泛化性理论、最优化理论，最后总结到支持向量机理论，并介绍了其实现技术和应用。本书的叙述循序渐进，内容深入浅出，既严谨又易于理解，得到了很多支持向量机研究者的认可。本书的原著已经是第二版第四次印刷。其一大特色是作者建立了一个网站 http://www.support-vector.net/，提供了在线的参考文献和一些软件的链接。这被评为“独特并具有现代化色彩”。目前国内还很缺乏这样系统性的著作，所以译者很乐意将本书翻译过来，供国内支持向量机、神经网络、机器学习、模式识别、数据挖掘乃至统计数学等领域的研究者参考。

译者在翻译过程中力求忠于原著。专业术语尽量遵循各学科的标准。由于水平和时间有限，对于原著的理解可能会有偏差，书中错误和不妥之处在所难免，恳请读者批评指正。

本书第 1 章和第 5 章初稿分别由曾华军和王猛翻译，其余章节由李国正翻译，全书由李国正整理完成。上海大学陈念贻教授校对了部分内容，并对一些翻译用语提出了中肯的建议；南京大学周志华教授就前六章细致地修改了一些错误的地方，并对专业术语的使用提出了中肯的意见；天津大学郑建华同学仔细阅读了译稿并提出了若干修改意见；清华大学孙建涛博士，北京的邹涛研究员，上海的叶晨周博士，交通大学的王永刚博士和青岛朗讯的刘玉海研究员分别对部分章节的翻译提出了一些宝贵的修改意见。南京航空航天大学陈松灿教授也对本书的翻译予以支持。译者谨在此对这些老师和同学表示感谢！李国正和王猛还要感谢导师杨杰教授提供的宽松舒适的学术环境以及在学习研究过程中所给予的指导。

前　言

近年来，支持向量机（SVM）的理论已经取得重大进展，其算法实现策略以及实际应用也发展迅速。可以确信，该技术的研究已发展成为机器学习中一个独立的子领域，在理论和实践两方面都有着光明的前景。尽管如此，目前还很缺乏对这个主题较系统的介绍材料。SVM 的理论体系涵盖的对象极为广泛，包括对偶表示、特征空间、学习理论、优化理论和算法等。

虽然关于这些主题仍在进行大量的研究工作，但其基础理论已经发展成熟，足以构建 SVM 的整个框架。本书从这些基础理论出发，尝试以 SVM 为主题进行深入浅出的介绍。

本书提供了严谨但易懂的阅读材料，可以作为学生或研究者进入机器学习这一新领域的指导读物。本书组织成教材形式，既可作为 SVM 课程的核心材料，也可作为神经网络、机器学习或模式识别等课程的附加读物。因为采取了教材形式，本书的内容尽可能做到自包含，以使非机器学习或者没有计算机背景的读者易于掌握。这样其他领域的读者就可以很轻松地应用 SVM 到自己的实际问题中去。作者还尝试通过严谨的推导来提供清晰的学习路线，因此书中提供了定理简要的证明过程以增强读者对概念的理解。对详细的证明过程感兴趣的读者可以参考原始文献。

每章后面附有习题，以及相关的参考文献或软件。由于在线资源可能会更新，所以有些引用会指向某一专门网站，其中相关的链接也会相应更新，从而保证读者找到相应的在线资源。

尽管有些引用是间接的，本书还是尽量标明所引用素材的作者。希望各位作者不会因为这些间接引用而感到不快。每章的结尾有补充读物和高级主题，其作用有二：一是将所有引用包含在这一节以使正文尽可能整洁，这里要再次请求所引用素材的作者宽恕这种延后的引用说明；二是希望为读者提供一个出发点以便能够进一步学习该章所讨论的主题。这些参考材料也包含在网站中，并保持更新。将引用说明移到正文之外的另一考虑是该领域已经到达一个成熟阶段，从而能使用前后一致的表述。但在这个方面有两个例外，一是某些定理通常可由其命名得知其作者，比如 Mercer 定理；另一例外是在第 8 章，其中描述了研究领域的一些特定实验。

本书写作中遵循的基本原则是对于不喜欢复杂的证明和概念的学生和研究者而言易于理解。相信通过直观且严谨的内容安排，SVM 的出现会显得简单又自然。本

书还尽量用简单的例子来介绍概念，之后才说明在复杂情形下的用法。本书是自包含的，附录中提供了基本线性代数和概率论之外的一些数学工具，从而更适合于跨学科的读者。

书中大部分材料在一个 5 小时的 SVM 和大间隔推广能力的讲座中展示过，该讲座是 1999 年在 California 大学 Santa Cruz 校区举办的。其中的大多数反馈意见也包含在书中。本书的部分章节是 Nello 在 California 大学 Santa Cruz 校区访问时写的，感谢这里的东道主和优美的校园环境。写作过程中，Nello 多次长时间访问 London 大学 Royal Holloway 校区。他很感谢 Lynda 及其家人的三次款待。Nello 和 John 还想感谢技术管理员 Alex Gammerman 以及 Royal Holloway 的计算机系同仁，他们提供了宽松舒适的环境，使得作者安心写作。

许多人为本书的雏形和主干做出了贡献，包括间接的讨论和对手稿的直接评注。作者感谢 Kristin Bennett, Colin Campbell, Nicolò Cesa-Bianchi, David Haussler, Ralf Herbrich, Ulrich Kockelkorn, John Platt, Tomaso Poggio, Bernhard Schölkopf, Alex Smola, Chris Watkins, Manfred Warmuth, Chris Williams 和 Bob Williamson。

作者还感谢 David Tranah 和 Cambridge 大学出版社对本书出版过程中的大力支持和帮助。

Alessio Cristianini 帮助建立了网站。Kostantinos Veropoulos 在 Bristol 大学用他的软件包生成了第 6 章的图片。感谢 John Platt 提供了附录 A 中的 SMO 伪码。

Nello 衷心感谢 EPSRC 的支持，感谢 Colin Campbell 主管的理解和协助。John 感谢欧洲委员会对于 NeuroCOLT2 工作组 EP27150 的支持。

由于第一版中出现了少量错误，作者尽力保证在重印前修改这些错误。欢迎读者指出更多的问题，并通过本书的网站 www.support-vector.net 反馈给作者。

Nello Cristianini 和 John Shawe-Taylor

2000 年 6 月

符 号 说 明

N	特征空间维数
$y \in Y$	输出和输出空间
$\mathbf{x} \in X$	输入和输入空间
F	特征空间
$\mathscr{F}$	实值函数类
$\mathscr{L}$	线性函数类
$\langle \mathbf{x} \cdot \mathbf{z} \rangle$	$\mathbf{x}$ 和 $\mathbf{z}$ 的内积
$\phi : X \to F$	到特征空间的映射
$K(\mathbf{x},\mathbf{z})$	核$\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$
$f(\mathbf{x})$	阈值化前的实值函数
n	输入空间的维数
R	包含数据的球半径
ε-insensitive	ε误差不敏感损失函数
$\mathbf{w}$	权重向量
b	偏置
$\boldsymbol{\alpha}$	对偶变量或者拉格朗日乘子
L	原拉格朗日函数
W	对偶拉格朗日函数
$\|\cdot\|_p$	p 阶范数
ln	自然对数
e	自然对数的底
log	以 2 为底的对数
$\mathbf{x}'$, $\mathbf{X}'$	向量、矩阵的转置
$\mathbb{N}$, $\mathbb{R}$	自然数、实数
S	训练样本
ℓ	训练样例个数
η	学习率
ε	误差概率

δ	置信范围
γ	间隔
ξ	松弛变量
d	VC 维

目　录

第1章 学习方法

长期以来，构造可以从经验中学习的机器在哲学界和科技界都是研究目标之一。从技术角度来讲，机器学习从电子计算机的发明中获得了强大的原动力。机器具有可观的学习能力，这一点已经得到了证实，然而这种学习能力的界定还远不够清晰。

开发可靠学习系统的重要性体现在有许多实际任务不能用传统的编程技术实现。目前并没有针对此问题的数学模型。比如，即使给予大量的样例，也不知道怎样用计算机程序来识别手写字符。因此很自然地产生疑问，能否训练计算机从样例中学习来识别字符“A”？毕竟这也是人类学习阅读的手段，可以把这种问题求解的途径称为学习方法。

同样的方法可适用于在DNA序列中寻找基因，过滤电子邮件，在机器视觉中检测或识别目标，等等。其中每个问题的解决都将为人类的生活带来革命性的变化，而每一个问题中机器学习都是其解决方案的关键所在。

本章将介绍学习方法的重要组成部分，总结各种不同种类的学习，并讨论其具有理论重要性的原因。在介绍学习方法的框架之后，给出了全书内容和关键问题的预览，并且解释了支持向量机为什么能够解决机器学习系统面临的问题。因此本章是对全书的预览，希望读者在阅读后面章节之前查阅本部分。

1.1 监督学习

当计算机应用到实际问题中时，通常可以显式地描述出给定一组输入如何推出所需的输出。而系统设计者和最终的程序实现人员的任务仅仅是将其转换为一系列的指令，使计算机能够遵循指令达到期望的结果。

由于计算机应用于更复杂的问题中，有时并不知道如何由给定输入计算出期望的输出，或者这种计算可能代价很高。例如对一个复杂的化学反应进行建模，则无法精确获知不同反应物质的相互作用关系；再如利用DNA序列对于蛋白质类型分类，或者对信用卡申请表的分类，以区分哪些人有能力偿还债务。

这些任务都不能用传统的编程途径来解决，因为系统设计者无法精确指定从输入数据得到输出的方法。解决此类问题的一种策略是让计算机从样例中学习输入到输出的函数对应关系，就像儿童学习辨认赛车的过程，是给他们大量赛车的例子，

而不是告诉他们赛车的精确规格说明。这种使用样例来合成计算机程序的过程称为学习方法，其中当样例是由输入/输出对给出时，称为监督学习。有关输入/输出函数关系的样例称为训练数据。

输入/输出对通常反映了把输入映射到输出的一种函数关系，然而有些情形下并非如此，比如输出中包含了噪声干扰. 当输入到输出存在内在函数时，该函数称为目标函数。由学习算法输出的对目标函数的估计称为学习问题的解。对于分类问题，该函数有时称为决策函数。存在一系列候选函数可把输入空间映射到输出域，可以选择其中之一为解。通常可以选择一组或一类候选函数，它们称为假设集合。例如，决策树是通过构造二叉树而产生假设，树的内部节点是简单的决策函数，而叶节点是输出值。因此把假设集合或者假设空间的选择看做学习过程的关键因素，而从训练数据中学习并从假设空间中选择假设的算法是第二个重要因素，它称为学习算法。

在学习区分赛车的例子中，输出为简单的是/否，它可看做是二元输出值。对于识别蛋白质类型的问题，输出值为有限数量的类别之一；对于化学反应的问题输出值为实数值表示的反应化合物的浓度。有二元输出的问题称为二类问题，有多个类别的问题称为多类问题，而实数值输出的问题称为回归问题。本书考察所有这些类型的学习问题，其中二类问题通常作为最简单情形被率先考虑。

其他还有一些学习类型不在本书中考虑。例如非监督学习，其数据不包含输出值，学习的任务是理解数据产生的过程。这种类型的学习包括密度估计、分布类型的学习和聚类等。还有一些学习模型考虑了学习器与其环境复杂的交互过程，其中最简单的情形是学习器可针对一特定输入向系统查询其输出。关于在该过程中如何影响学习器能力的研究称为查询学习。更复杂的交互在强化学习中考虑。这里学习器拥有一组可任意执行的动作，学习器通过执行这些动作尝试达到较高回报的状态。学习方法也可应用于强化学习中，前提是要将最优动作看做是学习器当前状态的一个函数输出。然而这样会产生相当的复杂性，因为输出的质量只能在动作的后果清晰后才能被间接衡量。

学习模型的另一方面的问题是训练数据如何生成及如何输入到学习器。例如，批量学习和在线学习之间存在明显的区别，前者在学习的一开始就把所有的数据提供给学习器；后者则让学习器一次只接收一个样例，并在接收正确输出前给出自己对输出的估计。在线学习中学习器根据每个新样例更新当前假设，学习器的质量由学习期间产生的总错误数量来衡量。

本书着重于在批量学习的设置下，根据有输出值的数据来学习输入/输出映射，即应用监督学习方法到批量训练数据上。

1.2 学习和泛化性

前面讨论了在线学习算法的质量可由其训练阶段的出错总数来衡量。然而，怎样衡量批量学习中所产生假设的质量还不明确。早期的学习算法致力于通过学习产生简单的符号表示。它可由专家来理解和验证。而当前情形下，学习的目标是输出一个假设以正确分类训练数据，早期的学习算法目标也是寻找对数据的精确拟合，这样寻找到的假设称为一致假设。然而生成可验证的一致假设这一目标存在两个问题。

一个问题是待学习的目标函数可能没有简单的表示，因此不能很容易地加以验证，例如在 DNA 序列中定位基因。某些子序列是基因，某些不是。但没有一种简单的方法来区分两者。

第二个问题是，通常训练数据是有噪声的。因此不能保证存在一个目标函数能够正确地映射训练数据。很明显信用检测是其中一例，因为偿债能力可能取决于其他一些系统无法获知的因素。另一个例子是网页分类的问题，这也是一个不精确的科学问题。

对于机器学习研究者来说，他们感兴趣的数据将越来越多地属于这种类型。因此使得上面的质量衡量标准难以实现。还有一个更为基本的问题在于，即使能够找到与训练数据一致的假设，它也可能无法对未见数据进行分类。一个假设正确分类训练集之外数据的能力称为泛化性，这正是要优化的属性。

把研究目标转移到泛化性，就不再要求把假设看做真实目标函数的正确表示。如果一个假设能给出正确的输出，它就满足泛化性准则，在这种意义上泛化性成为了一个功能性准则而不是描述性准则。同时该准则不再限制假设的规模以及“含义”，它们今后可以是任意的。

在后面可能看到这种重心改变被削弱的情形，因为可能需要搜索假设的简洁表示（即简短描述），这些可看做是有较好泛化性的属性。但目前，这种改变可看做是从符号表示到亚符号表示的转移。

第 4 章将给出这些概念的精确定义，那时将阐明使用这些模型的动机。

1.3 提高泛化性

泛化性准则对于学习算法附加了另一种约束。这一点可以由一种极端情形下的机械式学习来充分说明。许多经典的机器学习算法能够表示任意函数，并且对于困难的训练数据集会得到一个类似机械式学习器的假设。所谓机械式学习器是指能够

正确分类训练数据，但对所有未见数据会做出根本无关联性的预测。例如，决策树有可能过度增长直至针对每个训练样例有一叶子节点。为了得到一致假设而使假设变得过度复杂称为过拟合。控制此问题的一种方法是限制假设的规模，例如对于决策树可进行修剪操作。奥卡姆（Ockham）剃刀是该类方法中的准则之一，它建议如无必要，不必增加复杂性，或者说更精细的复杂性必须有利于显著提高训练数据的分类正确率。

这些观点自从奥卡姆的建议被采用后已有很长的历史。借此可引发复杂性和精度之间启发式的平衡，而且人们已提议了多种准则在这两者之间选择最优的折中。例如最小描述长度（MDL, Minimum Description Length）准则建议使用这样的假设：其函数描述长度与训练错误列表长度的和最短。

将要采用的方法是为了获得另一种平衡，它涉及泛化误差率上的统计边界，这些边界通常依赖于分类器间隔这样的变量，并引发最优化该变量的算法。该途径的缺点在于此算法不会好于统计结果。另一方面，算法的优点在于统计结果为其提供了一个有充分依据的理论基础，因此能避免基于错误直觉的启发式方法所带来的危险。

算法设计基于统计结果这一点并非意味着忽略解决此优化问题的计算复杂度。所感兴趣的技术需具有可伸缩性，它应能处理从玩具世界的问题到包含上万条记录的真实数据集的问题。只有通过对计算复杂度的原则性分析，才可能避免满足于那些只在小样本上表现良好，却对大训练集完全失效的启发式规则。计算复杂度理论研究了两类问题，第一类问题是是否存在算法能够在输入规模的多项式时间内运行的问题；第二类问题是如果存在这样的算法，任意解能否在多项式时间内检验，也就是能否在多项式时间内求解的问题。后一类问题即为 NP 完全问题，通常认为这些问题不能有效求解。

第 4 章将详细地阐述促使本书算法产生的统计结果。以下两种统计结果应加以区分：一种衡量了在给定训练样例有限时可获得的泛化性能；另一种是渐进统计结果，它研究了当样例数目趋于无穷时的泛化性能。本书要介绍的是前一种类型，该方法由 Vapnik 和 Chervonenkis 开创。

应当强调指出，还存在一些算法，它们与本书描述的不同，可由其他一些学习方法产生。在正文中将引用这些方法的相关文献。现在要稍微详细地介绍一下贝叶斯方法。

贝叶斯分析的出发点是假设集合上的先验分布，它描述了学习器对于数据特定假设的似然性的先验信念。只要能假定这样的先验分布，再加上数据如何被噪声干扰的模型，原则上就有可能在给定训练集合的情况下估计最可能的假设，甚至也可以在可能假设的集合上做加权平均。

如果不对所有的可能假设（即从输入空间到输出域的所有可能的目标函数）的集合加以限制，学习是不可能完成的。因为训练数据本身无法对未见样例进行分类。如果放宽限制，使得可以在看到数据之后再自由地选择假设集合，这同样也会产生问题，因为可能会简单地假定正确的假设具有任意先验概率。在此意义上所有学习系统必须做出贝叶斯模型的先验假定，它常称为学习偏置。第 4 章将在此背景下讨论本书所采用的方法。

1.4 学习的价值和缺点

容易理解的是，学习方法的前景是极为诱人的。首先，可由此途径解决的应用问题范围很广。其次，可望避免传统求解方法中烦琐的设计与编程，花费的代价只是收集一定数量的有输出值的数据，再运行一个现成的算法来学习输入/输出映射。最后，借此可以了解人类作用的内在方式，这一动机曾激励了神经网络的早期研究，当然它偶尔也会破坏科学的客观性。

然而，在学习方法中还存在许多难题需要仔细地研究和分析。例子之一是选择用于寻找输入/输出映射的函数类。此函数类必须具有足够的适应性以利于寻找所需的映射或它的近似。然而若此类过大，学习的复杂性会很高，特别是考虑到在大函数类中得到统计可靠的推理所需样例数目时更是如此。因此，在三层结点的神经网络中学习已知是 NP 完全问题，而最小化阈值线性函数的训练错误数目的问题也是 NP 难的。注意到这些难题后，会很清楚该方法的可应用性受到严格的限制，并且任何希望一举扫除所有困难的尝试都将是徒劳的。

在实践中，这些问题表现为一些特定的学习难题。首先，学习算法可能是低效的，比如出现局部最小值的情形。第二，输出的假设规模经常可能是大到不切实际。第三，如果训练样例数目有限，过大的假设函数类将导致过拟合以及很差的泛化性。第四个问题在于学习算法常常受到大量参数的控制，它们的选择往往是通过启发式的参数调节过程，使得系统的使用困难且不可靠。

尽管有这些缺陷，学习方法在实际问题上的应用还是取得了可观的成功。然而遗憾的是，还没有深刻理解成功的应用所需的条件，在算法及统计推理上都是如此。下一节将看到支持向量机能处理所有这些问题。

1.5 用于学习的支持向量机

支持向量机（SVM, Support Vector Machine）是在高维特征空间使用线性函数假设空间的学习系统，它由一个来自最优化理论的学习算法训练，该算法实现了一个

由统计学习理论导出的学习偏置。此学习策略由 Vapnik 和他的合作者提出，是一个准则性的并且强有力的方法。在它提出后的若干年来，在范围广大的应用中，SVM 的性能胜过其他大多数的学习系统。

本书提供了支持向量机的一个导论。第 2 章和第 3 章描述了假设空间及其表示，第 4 章学习偏置，第 5 章和第 7 章学习算法。第 6 章是关键的一章，它把所有这些部件总和在一起。而第 8 章提供了在现实问题上的某些应用的综述。本书以模块化方式组织，因此对某章内容熟悉的读者可以完全略过此章。特别要指出的是，如果读者希望直接获知何为 SVM 及如何实现它，可以直接阅读第 6 章和第 7 章。

第 2 章介绍了线性学习器，它是系统的主要构件。第 3 章介绍核函数，它用于定义隐式的特征空间，线性学习器可在其上工作。核函数是有效运用高维特征空间的关键。高维易产生过拟合的危险需要一个复杂的学习偏置，它是由第 4 章中统计学习理论的内容提供的。第 5 章涵盖了最优化理论，对解的属性给予了精确的刻画，从而指引在第 7 章设计出有效的学习算法，它还确保输出假设具有简洁的表示。对凸学习偏置的选择还完全避免了局部最小值的问题，因此即使对于包含上万条样例的训练集，总是可以找到有效的解。而假设的简洁表示意味着在新的输入上求值将非常迅速。因而，训练的效率、测试的效率、过拟合以及算法参数调节这四个困难都将在 SVM 中避免。

1.6 习题

1. 作为一个二分类问题，区分爬行动物和哺乳动物的过程是什么？输入空间是什么？为使计算机处理这样的数据，输入该如何表示？给出一个能实现该分类规则的函数的例子。
2. 对于下列类型的动物，重复习题 1：鸟类/鱼类；鱼类/哺乳类；哺乳类/鸟类。写出这四个决策规则中所使用的决策函数。
3. 给定一组正确标注的哺乳动物和鱼类：

   {狗、猫、海豚}，{金鱼、鲨鱼、金枪鱼}

   假设空间为习题 1 和习题 2 中获得的四个函数集合，如何选择正确的决策规则？

1.7 补充读物和高级主题

从数据中学习的问题在历史上已被许多哲学家研究过，并命名为“归纳推理”。

或许有些不可思议的是，直到20世纪人们才认识到，除非有先验知识，否则纯归纳是不可能的。这一概念性的成就基本上归功于Karl Popper[119]的基础性工作。

在统计的框架内研究该问题已有很长的历史。高斯在18世纪提出了最小均方回归的思想，而1930年Fisher的分类方法[40]仍是多数分析工作的出发点。

人工智能领域的研究者从一开始就考虑了学习的问题。Alan Turing [154]在1950年指出了学习器的思想，以反驳Lady Lovelace的“机器只会做我们指挥它们做的事”的论断。在此论文中还富有远见地提出了亚符号学习，Turing评论道：“学习器的一个重要特征是，其施教者对于内部实际的运转过程在很大程度上是无知的，而受教者的行为在一定程度上仍是可以预测的。”仅仅数年后，初始的学习器被开发出来，例如Arthur Samuel的跳棋程序[124]是强化学习的一个早期例子，而Frank Rosenblatt的感知机[122]包含了下一章将讨论的系统的许多特征。特别要指出，把学习问题建模使其成为适当假设空间中的搜索问题是人工智能方法的特点。Solomonoff还在其著名的论文[151,152]中研究了作为归纳推理的学习问题。

学习算法的发展成为人工智能的一个重要的子领域，最终形成了机器学习这样一个独立的学科领域。对机器学习中许多问题给予了“第一介绍”的一本可读性很强的书是Tom Mitchell的《机器学习》[99]。支持向量机由Vapnik及其合作者在其COLT论文[19]中提出，细节部分在Vapnik的著作[159]中描述。

这些参考文献也在网站www.support-vector.net上给出，这个网站将不断及时充实新的研究成果并提供在线软件和论文的链接。

第2章　线性学习器

对监督学习来说，学习器会输入一个带有标记（或者输出值）的样例（或输入）的训练集。样例通常是以属性向量的形式给出，因此输入空间是$\mathbb{R}^n$的子集。一旦给定输入向量，就可以为问题选择一定数目的假设函数集。其中，线性函数最容易理解而且应用最简单。传统统计学和经典神经网络文献已经阐述了许多利用线性函数区分两类事件的方法，以及利用线性函数插值的方法。包括有效的迭代方法及其泛化性能的理论分析等在内的技术提供了一个框架，以下各章将在这个框架内介绍更加复杂的系统。本章将回顾与支持向量机的研究相关的文献和结论。首先讨论分类的算法和相关主题，然后讨论与回归有关的问题。全书涉及到的学习器都将使用把输入变量的线性组合作为线性学习器的假设。

重要的是本章还将展示出，在大多数场合下这样的线性学习器可以表示成特定的有用形式，这种形式称为对偶表示。这种做法在以后的章节中将证明是很有用的。本章还将介绍间隔和间隔分布的重要符号。另外，在介绍分类问题的时候分类结果都是两类，在本章的最后将展示如何推广为多类。

2.1　线性分类

两类问题的分类通常用一个实值函数$f: X \subseteq \mathbb{R}^n \to \mathbb{R}$按照这样的方式操作：当$f(\mathbf{x}) \geqslant 0$时，输入$\mathbf{x} = (x_1, \ldots, x_n)'$赋给正类，否则赋给负类。考虑当$f(\mathbf{x}), \mathbf{x} \in X$是线性函数的情况，函数可以写为：

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \\ &= \sum_{i=1}^{n} w_i x_i + b \end{aligned}$$

这里$(\mathbf{w}, b) \in \mathbb{R}^n \times \mathbb{R}$是控制函数的参数，决策规则由$\operatorname{sgn}(f(\mathbf{x}))$给出，按照惯例，$\operatorname{sgn}(0) = 1$。学习方法意味着一定要从数据中学习到这些参数。

这类假设的几何解释是，式$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$定义的超平面将输入空间$X$分为两半（见图2.1）。超平面是维数为$n-1$的仿射子空间，它将空间分为两部分，这两部分对应输入中的两类。在图2.1中超平面是黑线，对应着上面的正区域和下面的负区域，

当 b 的值变化时，超平面平行于自身移动。因此，如果想表达$\mathbb{R}^n$中的所有可能超平面，包括$n+1$个可调参数的表达式是必要的。

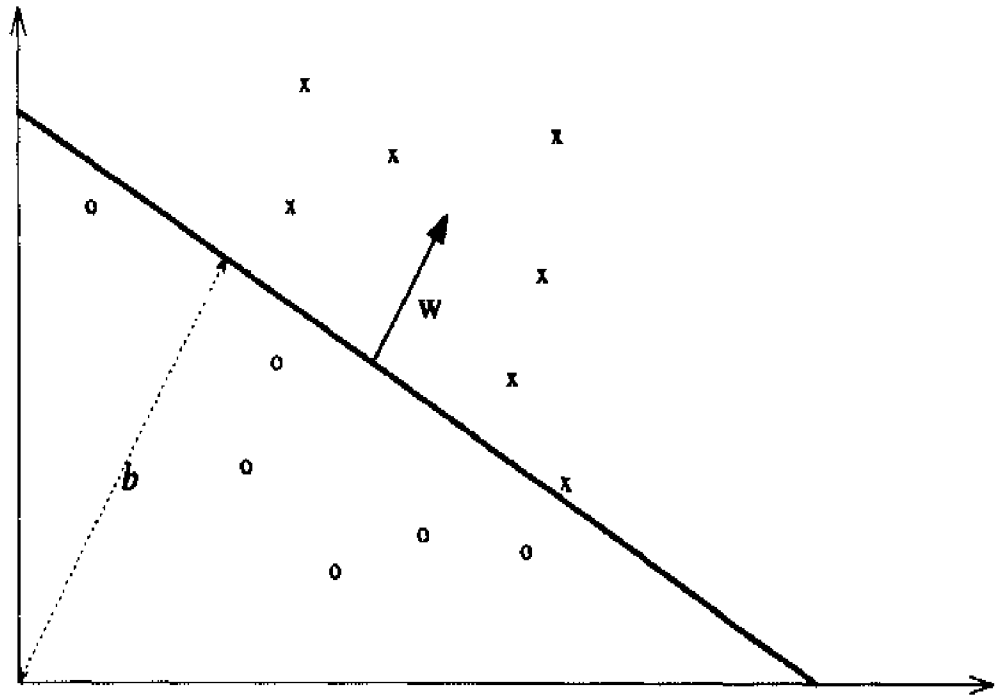

图 2.1　二维训练集的分开超平面（w,b）

统计学家和神经网络的研究者经常使用这种称为线性判别平面或感知机的简单分类器。线性判别平面的理论是 Fisher 在 1936 年发展起来的，而神经网络的研究者是在 20 世纪 60 年代早期开始研究感知机的，主要的工作由 Rosenblatt 完成。其中将 w和 b 称为权重向量和偏置，这是从神经网络文献中拿来的。有时候$-b$替代为θ，θ称为阈值。

在从样例中研究监督学习问题前，首先要介绍一些在全书中都要用到的符号，比如输入、输出、训练集，等等。

定义 2.1　一般使用 X 表示输入空间，Y 表示输出域。通常$X \subseteq \mathbb{R}^n$，对两类问题，$Y = \{-1,1\}$；对多类问题，$Y = \{1,2,\ldots,m\}$；对回归问题，$Y \subseteq \mathbb{R}$。训练集是训练样例的集合，训练样例也称为训练数据。通常表示为：

$$S = ((\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_\ell, y_\ell)) \subseteq (X \times Y)^\ell$$

其中，ℓ是样例数目。$\mathbf{x}_i$指样例，y_i是它们的标记。如果所有样例的标记相同，则训练集 S 是平凡的。注意，如果 X 是向量空间，输入向量就会成为像权重向量一样的列向量。如果想从$\mathbf{x}_i$中得到一个行向量，可以取转置$\mathbf{x}_i'$。

20 世纪 60 年代就已经提出了几个简单的迭代算法来优化代价函数，这些代价函

数使用超平面把点分为两类。下面的各小节将回顾一些最著名的算法，并且突出介绍它们的一些有趣的性质。感知机是有趣的，这不仅是因为历史原因，还因为在如此简单的一个系统内可以找到研究 SVM 理论所需要的绝大多数核心概念。注意有一些算法，比如最小二乘，既可用来做回归也可用来分类。为了避免重复，将在回归一节介绍。

2.1.1 Rosenblatt 感知机

线性分类器的第一个迭代学习算法是 Frank Rosenblatt 在 1956 年为感知机提出的。这个算法提出来以后，受到很大的关注。它是一个“在线”和“错误驱动”的程序，从一个初始权重向量 $\mathbf{w}_0$（通常 $\mathbf{w}_0 = \mathbf{0}$，一个全零的向量）开始，每次当一个训练点被现在的权重误分的时候都调整权重。该算法展示在表 2.1 中。这个算法直接更新权重向量和偏置，这里使用原始形式来跟下面将介绍的对偶表示形式做对比。

表 2.1 感知机算法（原始形式）

给定线性可分的数据集 S 和学习率 $\eta \in \mathbb{R}^+$
$\mathbf{w}_0 \leftarrow \mathbf{0}; b_0 \leftarrow 0; k \leftarrow 0$
$R \leftarrow \max_{1 \leqslant i \leqslant \ell} \|\mathbf{x}_i\|$
重复
  for $i = 1$ to ℓ
    if $y_i(\langle \mathbf{w}_k \cdot \mathbf{x}_i \rangle + b_k) \leqslant 0$ then
      $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \eta y_i \mathbf{x}_i$
      $b_{k+1} \leftarrow b_k + \eta y_i R^2$
      $k \leftarrow k + 1$
    end if
  end for
直到在 for 循环中没有错误发生
返回 $(\mathbf{w}_k, b_k)$，这里 k 是错误次数

如果存在一个超平面能够正确分类训练数据，并且这个程序保证收敛。这种情况称为线性可分。如果这样的超平面不存在，则数据称为不可分。

下面的定理展示迭代的次数与一个称为间隔的量相关。这个量在本书中占有重要地位，其正式的定义如下。

定义 2.2 样例 $(\mathbf{x}_i, y_i)$ 对应于超平面 $(\mathbf{w}, b)$ 的（函数的）间隔是量：

$$\gamma_i = y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b)$$

注意，$\gamma_i > 0$ 意味着 $(\mathbf{x}_i, y_i)$ 被正确分类。超平面 $(\mathbf{w}, b)$ 对应于训练集 S 的（函数的）间隔分布就是训练集 S 中样例的间隔分布。有时所谓间隔分布的最小值指超平面 $(\mathbf{w}, b)$

对应于训练集 S 的（函数的）间隔。

在两个定义中，如果把函数间隔替换为几何间隔，得到了跟归一化线性函数 $\left(\frac{1}{\|\mathbf{w}\|}\mathbf{w}, \frac{1}{\|\mathbf{w}\|}b\right)$ 等价的量，因而它度量了输入空间中的点到决策边界的欧氏距离。最终，训练集 S 的间隔是在所有超平面上的最大几何间隔。实现这个最大间隔的超平面称为最大间隔超平面。对于线性可分的训练集来说，间隔的值将都是正值。

图 2.2 显示了二维空间中两个点对应于超平面的几何间隔。当权重向量是单位向量时，这个几何间隔等价于函数间隔。图 2.3 显示了训练集的间隔。

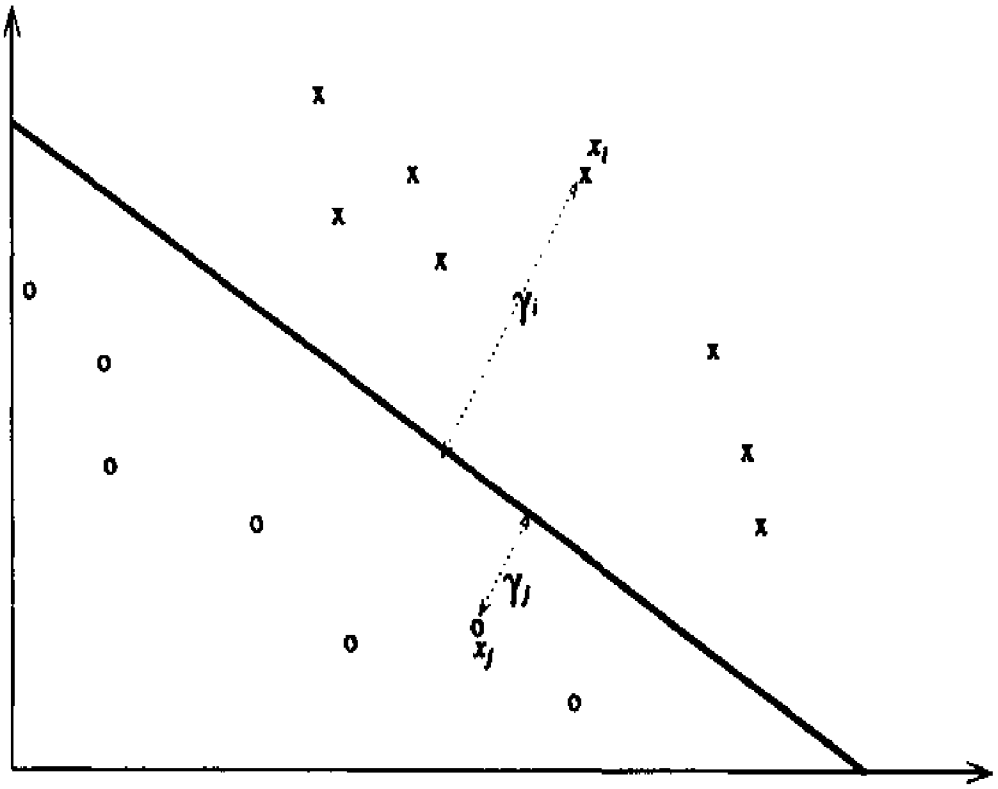

图 2.2　两个点的几何间隔

定理 2.3　（Novikoff）令 S 是一个非平凡的训练集，并且令：

$$R = \max_{1 \leqslant i \leqslant \ell} \|\mathbf{x}_i\|$$

假定存在向量 $\mathbf{w}_{\text{opt}}$，满足 $\|\mathbf{w}_{\text{opt}}\| = 1$ 并且对 $1 \leqslant i \leqslant \ell$ 有：

$$y_i(\langle \mathbf{w}_{\text{opt}} \cdot \mathbf{x}_i \rangle + b_{\text{opt}}) \geqslant \gamma$$

则 S 上在线感知机算法的误分次数最大为：

$$\left(\frac{2R}{\gamma}\right)^2$$

证明　为了便于分析，利用附加坐标 R 值扩充输入向量，新向量可以表示为 $\hat{\mathbf{x}}_i = (\mathbf{x}_i', R)'$，这里 $\mathbf{x}'$ 表示 $\mathbf{x}$ 的转置。类似地，可以结合偏置 b 增加附加坐标到权重向量 $\mathbf{w}$，得到扩充的权重向量 $\hat{\mathbf{w}} = (\mathbf{w}', b/R)'$。算法从扩充的权重向量 $\hat{\mathbf{w}}_0 = \mathbf{0}$ 开始，如果误分就更新权重值。令 $\hat{\mathbf{w}}_{t-1}$ 是在 t 个错误之前的扩充权重向量。那么，第 t 个更新如下进行：

$$y_i\langle\hat{\mathbf{w}}_{t-1}\cdot\hat{\mathbf{x}}_i\rangle = y_i(\langle\mathbf{w}_{t-1}\cdot\mathbf{x}_i\rangle + b_{t-1}) \leqslant 0$$

这里 $(\mathbf{x}_i, y_i) \in S$ 是被 $\hat{\mathbf{w}}_{t-1} = \left(\mathbf{w}_{t-1}', b_{t-1}/R\right)'$ 误分的点。更新按下面进行：

$$\hat{\mathbf{w}}_t = \left(\mathbf{w}_t', b_t/R\right)' = \left(\mathbf{w}_{t-1}', b_{t-1}/R\right)' + \eta y_i \left(\mathbf{x}_i', R\right)' = \hat{\mathbf{w}}_{t-1} + \eta y_i \hat{\mathbf{x}}_i$$

这里使用这个事实：

$$\begin{aligned} b_t/R &= b_{t-1}/R + \eta y_i R \\ \text{因为 } b_t &= b_{t-1} + \eta y_i R^2 \end{aligned}$$

下面的结论：

$$\langle\hat{\mathbf{w}}_t\cdot\hat{\mathbf{w}}_{\text{opt}}\rangle = \langle\hat{\mathbf{w}}_{t-1}\cdot\hat{\mathbf{w}}_{\text{opt}}\rangle + \eta y_i\langle\hat{\mathbf{x}}_i\cdot\hat{\mathbf{w}}_{\text{opt}}\rangle \geqslant \langle\hat{\mathbf{w}}_{t-1}\cdot\hat{\mathbf{w}}_{\text{opt}}\rangle + \eta\gamma$$

意味着（通过推导）可以得到：

$$\langle\hat{\mathbf{w}}_t\cdot\hat{\mathbf{w}}_{\text{opt}}\rangle \geqslant t\eta\gamma$$

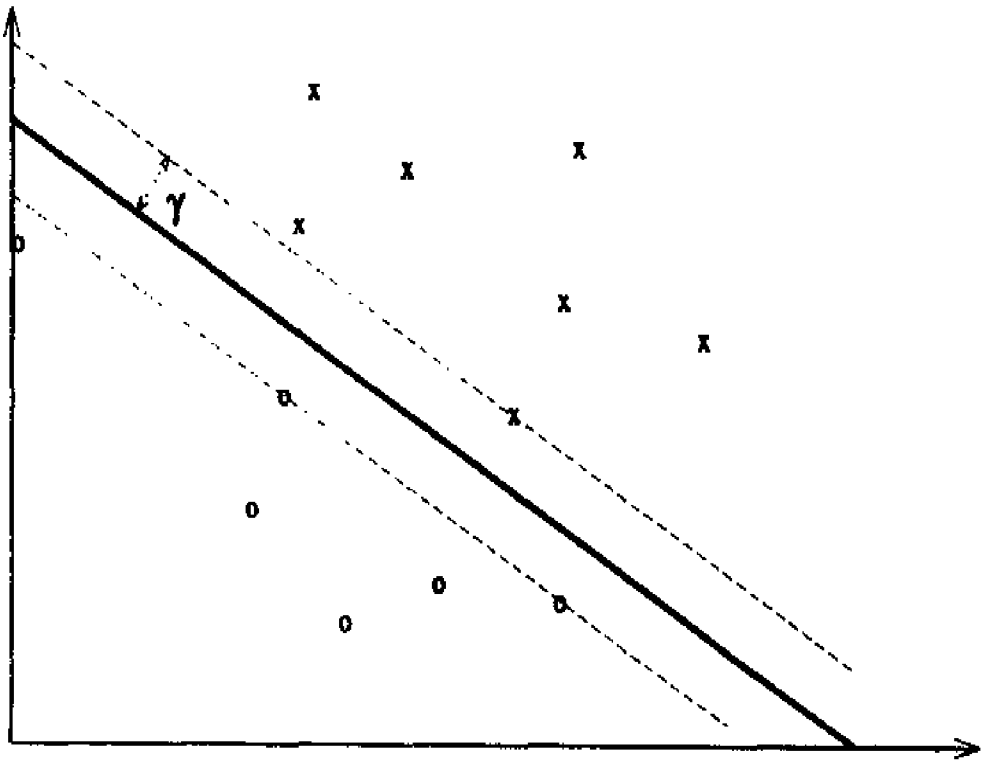

图 2.3　训练集的间隔

类似地有：

$$\begin{aligned}\|\hat{\mathbf{w}}_t\|^2 &= \|\hat{\mathbf{w}}_{t-1}\|^2 + 2\eta y_i\langle\hat{\mathbf{w}}_{t-1}\cdot\hat{\mathbf{x}}_i\rangle + \eta^2\|\hat{\mathbf{x}}_i\|^2 \\ &\leqslant \|\hat{\mathbf{w}}_{t-1}\|^2 + \eta^2\|\hat{\mathbf{x}}_i\|^2 \\ &\leqslant \|\hat{\mathbf{w}}_{t-1}\|^2 + \eta^2\left(\|\mathbf{x}_i\|^2 + R^2\right) \\ &\leqslant \|\hat{\mathbf{w}}_{t-1}\|^2 + 2\eta^2R^2\end{aligned}$$

这意味着：

$$\|\hat{\mathbf{w}}_t\|^2 \leqslant 2t\eta^2R^2$$

这两个不等式组合给出一个紧凑的关系：

$$\|\hat{\mathbf{w}}_{\text{opt}}\|\sqrt{2t}\eta R \geqslant \|\hat{\mathbf{w}}_{\text{opt}}\|\|\hat{\mathbf{w}}_t\| \geqslant \langle\hat{\mathbf{w}}_t.\hat{\mathbf{w}}_{\text{opt}}\rangle \geqslant t\eta\gamma$$

综合起来得到界：

$$t \leqslant 2\left(\frac{R}{\gamma}\right)^2\|\hat{\mathbf{w}}_{\text{opt}}\|^2 \leqslant \left(\frac{2R}{\gamma}\right)^2$$

既然对于数据的非平凡分开，有$b_{\text{opt}} \leqslant R$，那么有：

$$\|\hat{\mathbf{w}}_{\text{opt}}\|^2 \leqslant \|\mathbf{w}_{\text{opt}}\|^2 + 1 = 2$$

评注 2.4　这个定理通常假定零偏置，在这种情况下边界比一般情况下好 4 倍。然而在感知机算法中偏置会被更新，当使用标准更新算法（没有R^2因子），迭代次数与对应于扩充（包括偏置）权重的训练集的间隔有关。间隔总是小于或者等于γ，甚至更小。当$b_{\text{opt}} = 0$时，间隔等于γ，这时使用扩充间隔的界比一般情况下好 4 倍。当然在这种情况下上面证明的最后一行引入的 2 倍因子可以避免，使得界只相差 2 倍。相对而言，当$|b_{\text{opt}}| = O(R), R > 1$时，利用扩充权重的训练集获得的界是以$O(R^2)$因子比前面的界差。

评注 2.5　界中关键的量是包含数据的球的半径跟超平面的间隔的比的平方。直觉上认为这个量与学习率有关。但是这个比率在数据的正的尺度变换下是不变的，因此尺度变换不影响算法的迭代次数。原因是在阈值化线性函数的描述中有一个自由度，就是权重和偏置的正的尺度变换不改变分类结果。后面将使用这个结论来为可分数据集定义正则最大间隔超平面，这个超平面可通过将间隔固定为 1，最小化权重向量的范数来得到。所得范数值跟间隔成反比。

这个定理证明了当间隔是正的时候算法会在有限次数的迭代中收敛。如果分类超平面存在，仅需在序列 S 上迭代几次，在界为$\left(\frac{2R}{\gamma}\right)^2$的错误次数下，就可以找到分类超平面，算法停止。

在数据线性不可分的情况下，算法将不收敛：如果在序列 S 上执行迭代，算法将不停振荡，每当发生误分就改变假设$\mathbf{w}_i$。然而，存在类似 Novikoff 的定理给出了一次迭代中错误次数的界。它使用间隔分布的另一种度量方法，这种度量方法在以后的章节中很重要。直观地讲，它用的不是最靠近超平面的点，而是所有训练点所确定的间隔，这样就将间隔的概念进行了推广，从而可以说明训练样例更全局的特性。这个间隔分布的度量也可以用来度量样本的不可分性。

定义 2.6 固定$\gamma > 0$，定义样例$(\mathbf{x}_i, y_i)$对应于超平面$(\mathbf{w}, b)$和目标间隔γ的间隔松弛变量，为：

$$\xi((\mathbf{x}_i, y_i), (\mathbf{w}, b), \gamma) = \xi_i = \max\left(0, \gamma - y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right)\right)$$

这个量非正式地度量了一个点多大程度上没有以间隔γ离开超平面。当$\xi_i > \gamma$时，$\mathbf{x}_i$被$(\mathbf{w}, b)$误分。范数$\|\xi\|_2$度量了训练集中没有以间隔γ分开乃至误分的数目。

图 2.4 显示了两个误分点对应于单位范数超平面的间隔松弛变量。图中其他点的松弛变量为 0，因为它们的（正）间隔超过γ。

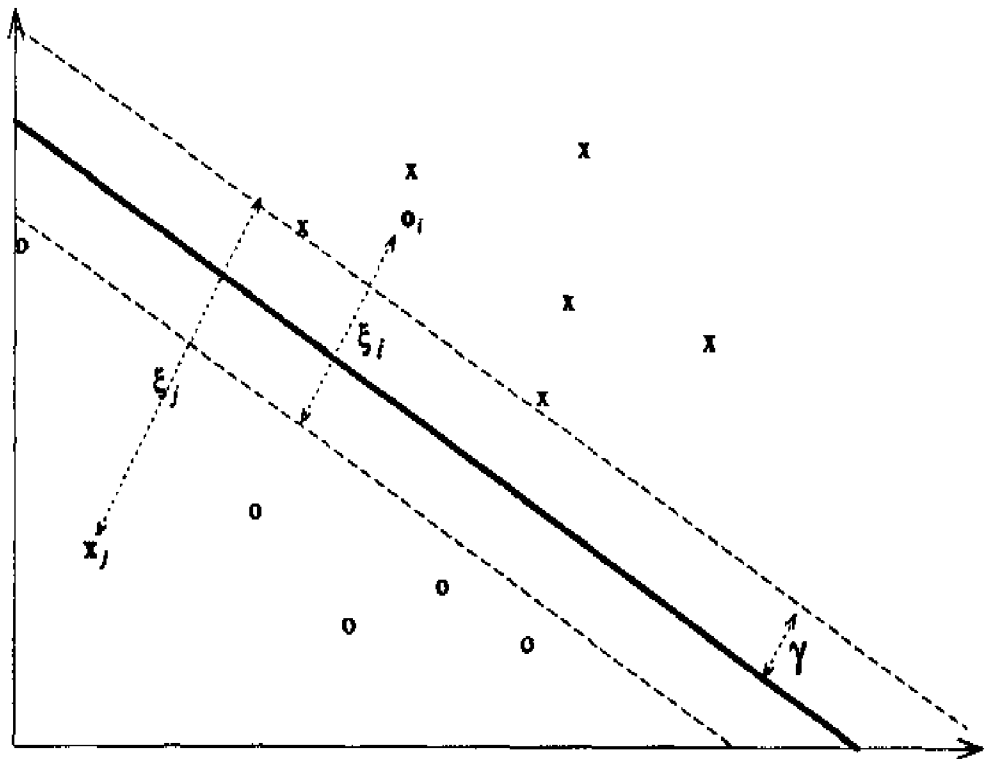

图 2.4 分类问题中的松弛变量

定理 2.7（Freund 和 Schapire）令 S 是没有重复样例的非平凡训练集，并且 $\|\mathbf{x}_i\| \leqslant R$。令 $(\mathbf{w}, b)$ 是任意 $\|\mathbf{w}\| = 1$ 的超平面。令 $\gamma > 0$ 并且定义：

$$D = \sqrt{\sum_{i=1}^{\ell} \xi_i^2} = \sqrt{\sum_{i=1}^{\ell} \xi\left((\mathbf{x}_i, y_i), (\mathbf{w}, b), \gamma\right)^2}$$

那么在数据集 S 上，表 2.1 感知机算法的 for 循环第一次执行的错误次数的界为：

$$\left(\frac{2(R+D)}{\gamma}\right)^2$$

证明　证明定义了一个 Δ 参数化的扩展输入空间，其中在未知数据上有一个功能相同的间隔 γ 和超平面 $(\mathbf{w}', b)'$。在扩展空间应用定理 2.3。最终，对 Δ 的选择优化得到了结果。扩展输入空间为每一个训练样例附加了一个坐标。样例 $\mathbf{x}_i$ 的项除去第 i 个附加坐标的值为 Δ 外，其他为零。令 $\tilde{\mathbf{x}}_i$ 表示扩展向量，$\tilde{S}$ 表示相应的数据集。赋值 $y_i\xi_i/\Delta$ 给 $\mathbf{w}$ 的第 i 个附加项得到扩展的向量 $\tilde{\mathbf{w}}$。可以发现：

$$y_i\left(\langle\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i\rangle + b\right) = y_i\left(\langle\mathbf{w} \cdot \mathbf{x}_i\rangle + b\right) + \xi_i \geqslant \gamma$$

显示，$(\tilde{\mathbf{w}}', b)'$ 在 $\tilde{S}$ 上有间隔 γ。注意，$\|\tilde{\mathbf{w}}\| = \sqrt{1 + D^2/\Delta^2}$，几何间隔 $\tilde{\gamma}$ 可以用这个因子简化。既然扩展的训练样例在不同坐标上有非零项，在 $\tilde{S}$ 上运行感知机算法的第一个 for 循环，同在 S 上运行有同样的效果，因此可以通过定理 2.3 给出错误次数的界：

$$\left(\frac{2\tilde{R}}{\tilde{\gamma}}\right)^2 = \frac{4\left(R^2 + \Delta^2\right)\left(1 + D^2/\Delta^2\right)}{\gamma^2}$$

这个界可以通过选择 $\Delta = \sqrt{RD}$ 进行优化从而得到所需结果。

评注 2.8　只可以将定理应用到 for 循环的第一个迭代的原因是当用一个训练样例 $\tilde{\mathbf{x}}_i$ 更新扩展空间的权重向量后，权重向量的第 i 个附加的坐标将有一个非零的项。当它在随后的迭代中处理的时候，非零项将影响 $\tilde{\mathbf{x}}_i$ 的评价。可以设想出一个感知机算法的变体以包含这些坐标，但 Δ 的值将成为一个参数，或者使对它的估计成为计算过程本身的一部分。

评注 2.9　尽管 D 可以定义到任意超平面，定理的界与线性可分的数据无关。以最小数目的误分来为非可分数据寻找一个线性分类面是一个 NP 完全问题。针对这个问题提出了一些启发式算法，比如口袋算法输出经过了最多迭代次数的 $\mathbf{w}$。前面的评注建议的扩展可以用来为非线性可分数据设计一个感知机算法。

重要的是要注意感知机算法通过在任意初始权重向量上增加误分的正训练样例

或者减去误分的负样例来工作。不失一般性，假定初始权重向量是零向量，这样最终的假设将是训练点的线性组合：

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$$

这里，既然 $\mathbf{x}_i$ 的系数由类别 y_i 决定，α_i 是个正值，正比于 $\mathbf{x}_i$ 被误分后权重更新的次数。这使得误分次数少的点将有较小的 α_i，而难分类的点将有较大的值。这个量有时候称为模式 $\mathbf{x}_i$ 的嵌入长度，在后面的章节中有重要地位。一旦样本 S 固定，向量 $\boldsymbol{\alpha}$ 是一个在不同或者对偶坐标上可供选择的表示假设的方法。展开式不是惟一的，不同的 $\boldsymbol{\alpha}$ 可能对应着相同的假设 $\mathbf{w}$。直观地讲，α_i 还可以看做 $\mathbf{x}_i$ 的信息量的指示。但在不可分数据的情况下，误分点的系数将无限增长。

在对偶坐标中，决策函数可以重写为：

$$\begin{aligned} h(\mathbf{x}) &= \operatorname{sgn}\left(\langle \mathbf{w} \cdot \mathbf{x} \rangle + b\right) \\ &= \operatorname{sgn}\left(\left\langle \sum_{j=1}^{\ell} \alpha_j y_j \mathbf{x}_j \cdot \mathbf{x} \right\rangle + b\right) \\ &= \operatorname{sgn}\left(\sum_{j=1}^{\ell} \alpha_j y_j \langle \mathbf{x}_j \cdot \mathbf{x} \rangle + b\right) \end{aligned} \tag{2.1}$$

并且这个感知机算法可以像表 2.2 一样全部用对偶形式表示。注意学习率仅仅改变超平面的尺度，不影响零起点向量的算法，所以不再包含它。

表 2.2 感知机算法（对偶形式）

给定训练集 S
$\boldsymbol{\alpha} \leftarrow \mathbf{0}$; $b \leftarrow 0$
$R \leftarrow \max_{1 \leqslant i \leqslant \ell} \|\mathbf{x}_i\|$
重复
  for $i = 1$ to ℓ
    if $y_i \left(\sum_{j=1}^{\ell} \alpha_j y_j \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle + b\right) \leqslant 0$ then
      $\alpha_i \leftarrow \alpha_i + 1$
      $b \leftarrow b + y_i R^2$
    end if
  end for
直到在 for 循环没有错误发生
返回定义式（2.1）中函数 $h(\mathbf{x})$ 的 $(\boldsymbol{\alpha}, b)$

感知机算法的替代形式和它的决策函数有许多有趣的性质。比如难学习的点有

较大的α_i，这一事实可用来对数据的信息含量排序。事实上，从简单感知机算法的分析中，已经发现了很多支持向量机理论将要用到的重要概念：间隔、间隔分布、对偶表示。

评注 2.10　既然更新的次数等于误分的数目，并且每次更新使得其中的成分加 1，向量α的一阶范数满足：

$$\|\alpha\|_1 \leqslant \left(\frac{2R}{\gamma}\right)^2$$

误分数目的界已在定理 2.3 给出。因而可以把α的一阶范数看做对偶表示中目标概念的复杂度。

评注 2.11　训练数据是以所知的 Gram 矩阵形式$\mathbf{G} = \left(\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle\right)_{i,j=1}^{\ell}$输入算法，Gram 矩阵的性质将在附录 B 中简要讨论，它还跟后面的其他矩阵有关联。这个发现的重要结果将在第 3 章论述。

2.1.2　其他线性分类器

学习一个超平面将两个（可分）点集分开的问题是一个不适定问题，在这个意义上，通常有若干个不同的解存在。比如在数据输入次序不同的情况下，感知机算法将产生不同的解。不适定问题的危险在于不是所有的解都是同样有用的。一种使其适定的方式是尝试用不同的代价函数优化，这样能保证得到的解是惟一的。比如，不是简单选择学习规则来正确分开两类，而是从这些规则中选择从数据产生最大距离的规则。这样的超平面可以实现最大间隔，也可以说有最大稳定性。和感知机算法相似的一个迭代算法可以保证收敛到最大间隔的解。本书将在第 7 章分析这种算法。

感知机算法仅在数据线性可分时保证收敛。不受此限制的一个方法是 Fisher 判别，它寻找数据投影分开最大的超平面$(\mathbf{w}, b)$。所优化的代价函数是：

$$F = \frac{(m_1 - m_{-1})^2}{\sigma_1^2 + \sigma_{-1}^2}$$

这里m_i和σ_i分别代表下面函数输出的平均值和标准偏差：

$$\{\langle \mathbf{w} \cdot \mathbf{x}_j \rangle + b : y_j = i\}$$

对两类，$i = 1, -1$。

优化这个准则的超平面可以通过求解一个带有对称矩阵的线性方程组来得到，

对称矩阵由训练数据形成，并且方程组右侧是两个类的平均值的差。

2.1.3 多类判别

对两类判别问题本章已经做了较多探讨，对其也可以通过为每个类定义一个权重向量$\mathbf{w}_i$和一个偏置b_i来解决。每次一个新样例需要分类的时候，两个函数都要评价，如果$\langle \mathbf{w}_1 \cdot \mathbf{x} \rangle + b_1 \geqslant \langle \mathbf{w}_{-1} \cdot \mathbf{x} \rangle + b_{-1}$，点$\mathbf{x}$赋予类1，否则赋予类-1。这个方法等价于使用单个超平面$(\mathbf{w}, b)$来判别，其中$\mathbf{w} = \mathbf{w}_1 - \mathbf{w}_{-1}$, $b = b_1 - b_{-1}$。

对于多类分类问题，输出域是$Y = \{1, 2, \ldots, m\}$。线性学习器推广到m类是很直接的：对m类中的每个类关联一个权重向量和一个偏置，$(\mathbf{w}_i, b_i)$，$i \in \{1, \ldots, m\}$，则决策函数给出如下：

$$c(\mathbf{x}) = \arg \max_{1 \leqslant i \leqslant m} \left(\langle \mathbf{w}_i \cdot \mathbf{x} \rangle + b_i \right)$$

从几何角度，这等价于给每个类关联一个超平面，然后将新点$\mathbf{x}$赋予超平面离其最远的一类。输入空间分为m个简单相连的凸区域。

从数据中同时学习m个超平面的算法由此得出，即上面列出的基本算法的扩展。

2.2 线性回归

线性回归的问题就是求线性函数：

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$$

使其能够最好地拟合一个给定标记为$Y \subseteq \mathbb{R}$的训练点集S。从几何角度讲是寻找一个拟合给定点的超平面。图2.5显示了一维线性回归函数。图中显示为ξ的距离是某个训练样例的误差。

这个问题从18世纪开始研究，最著名的解是Gauss和Legendre分别提出的寻找到训练点的误差平方和最小的直线的方法。这也就是最小二乘法，它能在线性目标被高斯噪声干扰的情况下获得最优结果。

数值稳定性和泛化性的考虑促使介绍该技术的一个改进，它类似分类情况下的最大间隔超平面：选择一个使得误差平方和权重向量$\mathbf{w}$的范数最小的函数。这个由Hoerl和Kennard提出的解就是岭回归。这些算法需要矩阵的逆，可以用一个简单的迭代程序实现（Widrow和Hoff在20世纪60年代提出的Adaline算法）。注意这些回归技术也能在分类问题中使用，当然必须要小心地选择和类关联的目标值。

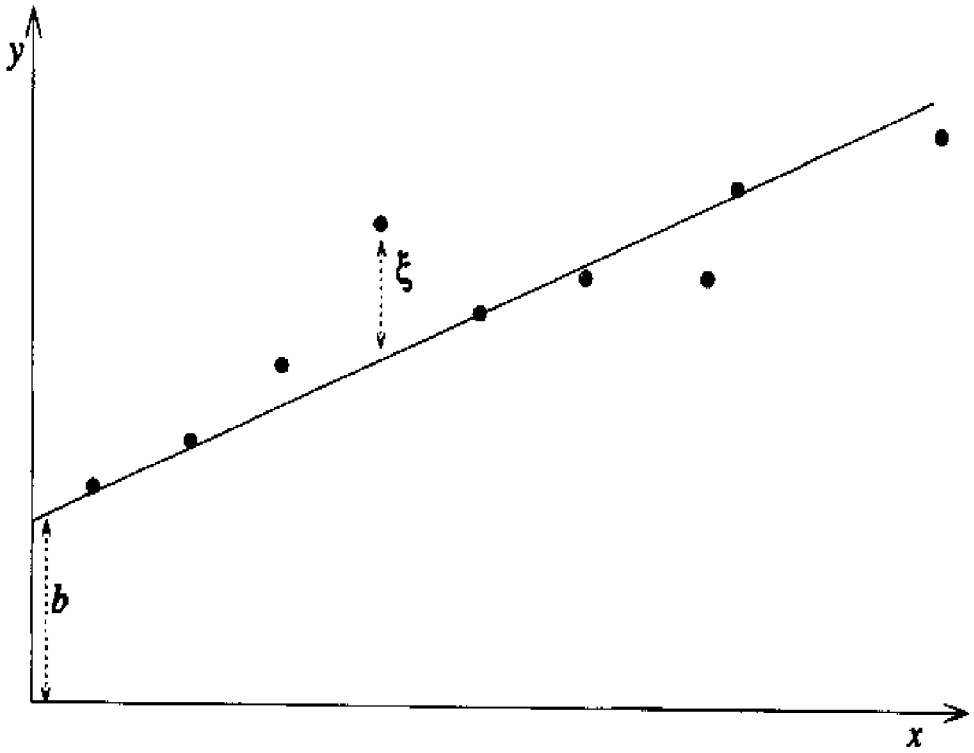

图 2.5　一维线性回归函数

2.2.1　最小二乘法

给定一个训练集 S，其中 $\mathbf{x}_i \in X \subseteq \mathbb{R}^n$，$y_i \in Y \subseteq \mathbb{R}$，线性回归的问题是寻找（线性）函数 f 来对数据建模：

$$y = f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$$

最小二乘的方法通过最小化偏离数据的误差平方和来选择参数 $(\mathbf{w}, b)$：

$$L(\mathbf{w}, b) = \sum_{i=1}^{\ell} \left(y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b \right)^2$$

函数 L 就是平方损失函数，它通过平方和度量了特定参数选择带来的损失。损失当然也可以通过其他损失函数度量。一定不要把符号 L 和第 5 章的拉格朗日函数混淆。可以通过对应于参数 $(\mathbf{w}, b)$ 求偏导并且令所得的 $n+1$ 个线性表达式为 0 来最小化 L。最好的表达方式是用矩阵，令 $\hat{\mathbf{w}} = (\mathbf{w}', b)'$，并且：

$$\hat{\mathbf{X}} = \begin{pmatrix} \hat{\mathbf{x}}_1' \\ \hat{\mathbf{x}}_2' \\ \vdots \\ \hat{\mathbf{x}}_\ell' \end{pmatrix} \text{ 这里 } \hat{\mathbf{x}}_i = (\mathbf{x}_i', 1)'$$

利用这个符号，输出的向量差成为：

$$\mathbf{y}-\hat{\mathbf{X}}\hat{\mathbf{w}}$$

其中y是一个列向量。因此损失函数写做：

$$L(\hat{\mathbf{w}})=(\mathbf{y}-\hat{\mathbf{X}}\hat{\mathbf{w}})'(\mathbf{y}-\hat{\mathbf{X}}\hat{\mathbf{w}})$$

取损失函数的导数，设为0：

$$\frac{\partial L}{\partial \hat{\mathbf{w}}}=-2\hat{\mathbf{X}}'\mathbf{y}+2\hat{\mathbf{X}}'\hat{\mathbf{X}}\hat{\mathbf{w}}=\mathbf{0}$$

得到著名的“标准方程”：

$$\hat{\mathbf{X}}'\hat{\mathbf{X}}\hat{\mathbf{w}}=\hat{\mathbf{X}}'\mathbf{y}$$

其中，如果$\hat{\mathbf{X}}'\hat{\mathbf{X}}$的逆存在，最小二乘问题的解是：

$$\hat{\mathbf{w}}=(\hat{\mathbf{X}}'\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}'\mathbf{y}$$

如果$\hat{\mathbf{X}}'\hat{\mathbf{X}}$是奇异的，可以使用伪逆，或者是下面描述的岭回归技术。

在20世纪60年代，注意力主要放在如何构造简单的迭代程序来训练线性学习器。Widrow-Hoff算法（也就是Adaline算法）能收敛到最小二乘解，和感知机算法相似，它实现了一个简单的梯度下降策略。算法显示在表2.3中。

表2.3 Widrow-Hoff算法（原始形式）

给定训练集S和学习率$\eta\in\mathbb{R}^+$

$\mathbf{w}_0\leftarrow\mathbf{0};\ b\leftarrow 0$

重复

　　for $i=1$ to ℓ

　　$(\mathbf{w},b)\leftarrow(\mathbf{w},b)-\eta\left(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b-y_i\right)(\mathbf{x}_i,1)$

　　end for

直到收敛条件被满足

返回$(\mathbf{w},b)$

2.2.2 岭回归

如果在最小二乘法中$\hat{\mathbf{X}}'\hat{\mathbf{X}}$矩阵不是满秩，或者在其他情况下数值解不稳定，则可以使用下面的解：

$$\hat{\mathbf{w}}=(\hat{\mathbf{X}}'\hat{\mathbf{X}}+\lambda\mathbf{I}_n)^{-1}\hat{\mathbf{X}}'\mathbf{y}$$

上面的公式可以通过在矩阵$\hat{\mathbf{X}}'\hat{\mathbf{X}}$上增加对角矩阵$\mathbf{I}_n$的乘子$\lambda\in\mathbb{R}^+$来得到，这里$\mathbf{I}_n$是一个$(n+1,n+1)$项为零的单位矩阵。这个解称为岭回归，最初是在统计领域因为数值

稳定性的原因提出的。

岭回归算法通过最小化惩罚损失函数：

$$L(\mathbf{w},b)=\lambda\langle\mathbf{w}\cdot\mathbf{w}\rangle+\sum_{i=1}^{\ell}\left(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b-y_i\right)^2 \tag{2.2}$$

使得参数λ在最小化解的范数与最小化损失平方和之间起到平衡作用。

评注 2.12　这个算法跟分类中的最大间隔算法相似，提出了一个复杂的代价函数。这个函数由两部分组成，一个控制假设的复杂度，另一个控制训练数据的精度。第 4 章将对这种代价函数及相关的线性学习器的泛化性能做系统研究。

注意岭回归算法允许对偶表示。它的解需要满足$\frac{\partial L}{\partial \mathbf{w}}=\mathbf{0}$，并为假设给出了下面的表达式：$\lambda\mathbf{w}=-\sum_i(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b-y_i)\mathbf{x}_i$，这意味着存在标量$\alpha_i=-\frac{1}{\lambda}\left(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b-y_i\right)$，使解可以写为$\mathbf{w}=\sum_i\alpha_i\mathbf{x}_i$。

一旦知道解可以表示为对偶形式，就可以导出$\boldsymbol{\alpha}$必须满足的条件。可以用向量表示对偶条件，权重向量可以用$\boldsymbol{\alpha}$写为：

$$\mathbf{w}=\mathbf{X}'\boldsymbol{\alpha}$$

这里$\mathbf{X}$是将矩阵$\hat{\mathbf{X}}$的最后列（1）去除得到。重写式（2.2）如下，这里为了简化，令b为 0：

$$\begin{aligned}L(\mathbf{w}) &= \lambda\boldsymbol{\alpha}'\mathbf{X}\mathbf{X}'\boldsymbol{\alpha}+\sum_{i=1}^{\ell}\left(\boldsymbol{\alpha}'\mathbf{X}\mathbf{x}_i-y_i\right)^2\\ &= \lambda\boldsymbol{\alpha}'\mathbf{G}\boldsymbol{\alpha}+\sum_{i=1}^{\ell}\left((\mathbf{G}\boldsymbol{\alpha})_i-y_i\right)^2\\ &= \lambda\boldsymbol{\alpha}'\mathbf{G}\boldsymbol{\alpha}+(\mathbf{G}\boldsymbol{\alpha}-\mathbf{y})'(\mathbf{G}\boldsymbol{\alpha}-\mathbf{y})\\ &= \lambda\boldsymbol{\alpha}'\mathbf{G}\boldsymbol{\alpha}+\boldsymbol{\alpha}'\mathbf{G}\mathbf{G}\boldsymbol{\alpha}-2\mathbf{y}'\mathbf{G}\boldsymbol{\alpha}+\mathbf{y}'\mathbf{y}\end{aligned}$$

这里$\mathbf{G}=\mathbf{X}\mathbf{X}'=\mathbf{G}'$。取对应于$\boldsymbol{\alpha}$的导数，令其为零，得到下面的方程：

$$2\mathbf{G}(\lambda\boldsymbol{\alpha}+\mathbf{G}\boldsymbol{\alpha}-\mathbf{y})=\mathbf{0}$$

要使上式成立，需要满足条件：

$$(\lambda\mathbf{I}+\mathbf{G})\boldsymbol{\alpha}=\mathbf{y}$$

它给出了一个预测函数：

$$f(\mathbf{x})=\mathbf{y}'(\lambda\mathbf{I}+\mathbf{G})^{-1}\mathbf{z}$$

此处$z_i=\langle\mathbf{x}\cdot\mathbf{x}_i\rangle$。注意这个对偶方程与训练样例内积的 Gram 矩阵即$\mathbf{G}=\mathbf{X}\mathbf{X}'$有关。

2.3 线性学习器的对偶表示

前一节主要强调多数线性学习器都存在对偶表示形式。这个表示形式将在下面的章节使用，并成为一类算法的通用性质。对偶性是支持向量机的关键概念之一。

对偶表示的一个重要性质是数据仅作为 Gram 矩阵的项出现，而不需要通过单个属性出现。类似地，在决策函数的对偶表示里仅需要与测试点数据的内积。这个事实将在本书剩余的章节产生重要影响。

最后要注意的是在第 5 章将系统地研究本章经验性得到的关于对偶性的课题。这里讨论的许多问题和算法将作为最优化问题的特殊情况，因为最优化问题有一个自然包容对偶性的数学框架。

2.4 习题

1. 写出 Widrow-Hoff 算法的对偶形式。
2. 为线性回归算法用原始形式和对偶形式写一个迭代算法。
3. 使用评注 2.8 和评注 2.9 中的思路为不可分数据开发一个感知机算法。

2.5 补充读物和高级主题

线性判别理论的出现可以追溯到 20 世纪 30 年代，当时 Fisher[40]提出了一个分类算法。在人工智能领域该理论因 Frank Rosenblatt[122]的工作而受到关注，Frank 从 1956 年开始介绍感知机算法的规则。Minsky 和 Papert 的名著《感知机》[98]分析了线性学习器的局限。Duda 和 Hart 的名著[35]提供了截至 1973 年的完整的最新成就的综述。更多的成果可以参考[16]，其中包括了一类通用的学习器的描述。

将 Novikoff 理论[104]扩展到不可分情况归功于[43]；Gallant[47]提出了口袋算法；[35]对 Fisher 判别做了简单描述。对于在不可分情况下线性分类的计算复杂性的讨论，请见[64]和[8]。最大间隔超平面的思想出现过多次，Vapnik 和 Lerner 在[166]中讨论过，Duda 和 Hart 在[35]中讨论过，在统计类的文献[4]中为最大间隔超平面提出了一个 *Adatron* 的迭代算法，这个算法将在第 7 章中进一步讨论。

线性回归的问题比分类问题更古老，最小二乘线性插值由 Gauss 在 18 世纪为天文问题首先提出。Hoerl 和 Kennard[63]发表了岭回归算法，然后作为 Tikhonov[153]解决不适定问题的正则化理论的一个特例进一步提出。岭回归算法的对偶形式包括第 2.2.2 小节导出的结论，在 Saunders 等的[125]和[144]中研究过。一个等价的启发

式规则在神经网络文献中以权重衰减的名字被广泛采用。Widrow-Hoff 算法则在[179]中描述。

最后，要注意的是线性学习器使用训练数据的对偶表示，它与拉格朗日乘子的优化技术关系密切，将在第 5 章进一步讨论。Guyon 和 Stork[56]比较了线性学习器原始形式和对偶形式的表示，这种对偶形式跟本章采用的形式相似。

这些参考文献也在网站 www.support-vector.net 上给出，这个网站将不断及时充实新的研究成果并提供在线软件和论文的链接。

第3章　核函数特征空间

Minsky 和 Papert 在 20 世纪 60 年代明确指出线性学习器计算能力有限。总体上，现实世界复杂的应用需要有比线性函数更富有表达能力的假设空间。换言之，就是目标概念通常不能由给定属性的简单线性函数组合产生，而是应该一般地寻找待研究数据的更抽象的特征。多层阈值线性函数可以作为这个问题的一个解，由此导向了多层神经网络和训练该学习系统的后向传播算法。

核表示方式提供了另一条解决途径，即将数据映射到高维空间来增加第2章中线性学习器的计算能力。线性学习器对偶空间的表达方式使得这个步骤的隐式操作成为可能。第2章已经评注过，训练样例不会独立出现，而总是以成对样例的内积形式出现。用对偶形式表示学习器的优势在于在该表示中可调参数的个数不依赖输入属性的个数。通过选择使用恰当的核函数来替代内积，可以隐式地将训练数据非线性映射到高维空间，而不增加可调参数的个数，当然前提是核函数能够计算对应着两个输入特征向量的内积。

本章将介绍核算法，它为支持向量机（SVM）提供了一个重要的构成模块。SVM 的一个重要特征就是在一定程度上逼近理论的问题与学习理论的问题相互独立。因此，可以用一种通用的和自包含的方式研究核表示方式的性质，并且在不同的学习理论中使用它们，比如在第 4 章。

核方法的另一个吸引人的地方是学习算法和理论可以很大程度上同应用领域的特性分开，这些特性可以在设计合适核函数时加以考虑。因此，在神经网络应用中结构选择的问题，在 SVM 的应用中成为选择合适核函数的问题。本章将描述几种著名的核函数，并且展示如何利用简单的核函数来构造复杂的核函数。其中还将提到为文本等离散结构所开发的核函数，显示了这个方法的输入空间不局限在欧氏空间，而是可以应用到不能定义线性函数的输入空间。

像将在第4章和第7章介绍的一样，在计算和泛化性方面核函数的使用都能够克服维数灾难。

3.1　特征空间中的学习

需要学习的目标函数的复杂度取决于它的表示方式，学习任务的难度也随之变

化。理想情况下，应该选择与特定的学习问题匹配的表示。因此，在机器学习中一个普通的预处理策略包括改变数据的表达形式：

$$\mathbf{x} = (x_1, \ldots, x_n) \longmapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x}))$$

这个步骤等价于将输入空间 X 映射到一个新的空间，$F = \{\boldsymbol{\phi}(\mathbf{x}) | \mathbf{x} \in X\}$。

例 3.1　目标函数：

$$f(m_1, m_2, r) = C\frac{m_1 m_2}{r^2}$$

给出牛顿万有引力定律，用两个物体质量 m_1，m_2 和距离 r 表示万有引力。这个定律用可观测的量，即质量和距离来表示。第 2 章介绍的线性学习器就不能表达这种关系，但若简单地变换坐标：

$$(m_1, m_2, r) \longmapsto (x, y, z) = (\ln m_1, \ln m_2, \ln r)$$

给出表达式：

$$g(x, y, z) = \ln f(m_1, m_2, r) = \ln C + \ln m_1 + \ln m_2 - 2\ln r = c + x + y - 2z$$

就可以通过一个线性学习器学习。

将数据简单映射到另一个空间能够很好地简化任务，这个事实在机器学习中很早就发现了，并且给出了很多选择数据最优表达形式的技术。描述数据的量，通常称为特征，而原始的量有时称为属性。选择最合适表达式的任务称为特征选择。空间 X 是指输入空间，而 $F = \{\boldsymbol{\phi}(\mathbf{x}) : \mathbf{x} \in X\}$ 则称为特征空间。

图 3.1 展示了特征从二维输入空间映射到二维特征空间的例子。在输入空间数据不能通过线性函数分开，但在特征空间是可以的。本章的目的是展示如何映射到高维空间使得线性分类更容易。

已经存在多种不同的特征选择方法。通常要寻找包含了原始属性中必要信息的最小特征集，即所谓维数约简：

$$\mathbf{x} = (x_1, \ldots, x_n) \longmapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_d(\mathbf{x})) \quad d < n$$

它对计算和泛化性能都有益，因为这两者随着特征的增加性能会下降，即所谓维数灾难。高维特征空间的困难是很不幸的，因为（可能冗余的）特征集越大，使用标准学习器表示的函数就能拟合得越好。本书将展示 SVM 如何避免这种性能的下降。

例 3.2　进一步考虑两个物体间的万有引力定律。假定属性是与位置有关的三个成分，再加上两个质量：

$$\mathbf{x} = (p_1^x, p_1^y, p_1^z, p_2^x, p_2^y, p_2^z, m_1, m_2)$$

一种约简维数的方法是下面的映射$\phi : \mathbb{R}^8 \longmapsto \mathbb{R}^3$：

$$\mathbf{x} = (p_1^x, p_1^y, p_1^z, p_2^x, p_2^y, p_2^z, m_1, m_2) \mapsto \phi(\mathbf{x}) = \left(\sqrt{\sum_{i \in \{x,y,z\}} (p_1^i - p_2^i)^2}, m_1, m_2 \right)$$

这能保持必要的信息。

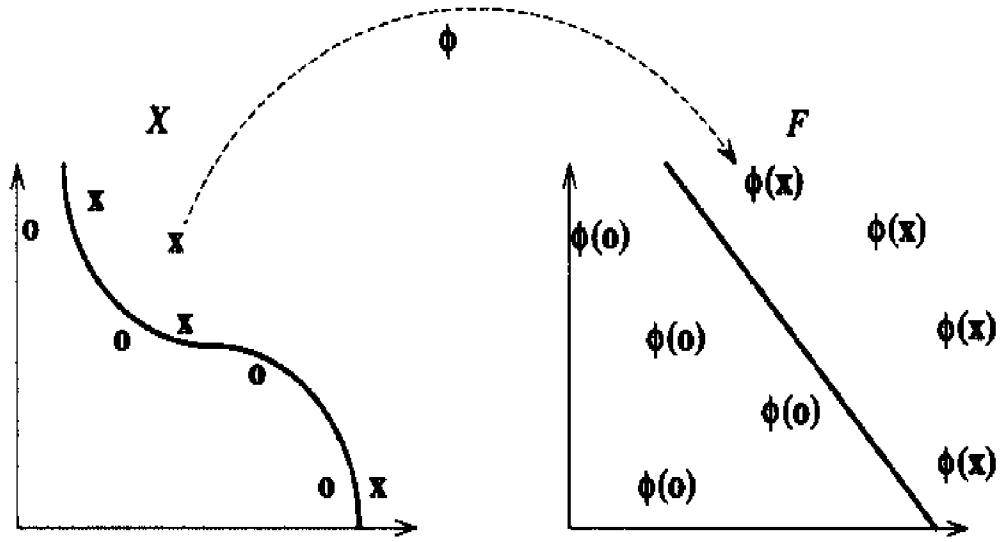

图 3.1 简化分类任务的特征映射

另一个不同的特征选择任务是检测出无关特征并将其去除。在本章的例子中，无关的特征可以是物体的颜色或者是温度，因为这两个量同目标值输出无关。主成分分析提供了一种将数据映射到特征空间的方法，它将原始属性线性组合，并根据数据展示在每个特征方向的方差大小排序。维数约简有时会简单地去除那些方差很小的方向上的特征，尽管不能保证这些特征对于目标分类不重要。下面给出一个额外特征有用的例子。

例 3.3 考虑二维输入空间的情况，假定关子问题的先验知识提示相关信息已经编码到自由度为 2 的单项式的形式。因此，要试图将问题在一个特征空间中表示，并且在空间中展示出相关信息，准备好为学习器所用，一个可能的映射是：

$$(x_1, x_2) \longmapsto \phi(x_1, x_2) = (x_1^2, x_2^2, x_1 x_2)$$

同样的方式，我们可能想使用自由度为 d 的特征，给出$\binom{n+d-1}{d}$维的特征空间，但即使不太大的属性数目和特征自由度，特征空间维数很快就变得不可计算。使用这种类型的特征空间需要一种特殊的技术，该技术将在第 3.2 节介绍，它是将数据隐式映射到特征空间。

计算问题不是惟一因特征空间太大而导致的问题。另一个问题是学习器的泛化能力，它对假设的标准函数类表示方式的维数很敏感。从前面的例子中可以很明显地看出特征选择是学习过程的一个部分，应该尽可能使其自动化。

从另一方面来说，它又是较任意的一步，反应了对潜在目标函数的先验期望。学习的理论模型应该考虑到这个问题：除非泛化能力可以某种方式控制，使用过大的特征集会引起过拟合问题。这就是维数约简技术受到重视的原因。然而，在第 4 章可以看到，对泛化性的深入理解表明，甚至可以使用无限维的特征空间。泛化性的问题可以通过使用在这个理解基础上的学习器来避免，计算问题也可以通过下一节描述的隐式映射来避免。

3.2　到特征空间的隐式映射

为了用线性学习器学习一个非线性关系，需要选择一个非线性特征集，并且将数据写成新的表达形式。这等价于应用一个固定的非线性映射，将数据映射到特征空间，在特征空间中使用线性学习器。因此，考虑的假设集是这种类型的函数：

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x}) + b$$

这里 $\phi : X \to F$ 是从输入空间到某个特征空间的映射。这意味着建立非线性学习器分为两步：首先使用一个非线性映射将数据变换到一个特征空间 F，然后在这个特征空间使用线性学习器分类。

就像在第 2 章显示的一样，线性学习器的一个重要性质是可以表达为对偶形式。这意味着假设可以表达为训练点的线性组合，因此决策规则可以用测试点和训练点的内积来表示：

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle + b$$

如果有一种方式可以在特征空间中直接计算内积 $\langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle$，就像在原始输入点的函数中一样，就有可能将两个步骤融合到一起建立一个非线性的学习器。这样直接计算的方法称为核函数方法。

定义 3.4　核是一个函数 K，对所有 $\mathbf{x}, \mathbf{z} \in X$，满足：

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$$

这里 ϕ 是从 X 到（内积）特征空间 F 的映射。

核这个名字是从积分算子理论中来的，这个理论以核与其相关特征空间的关系的理论为基础。对偶表达的一个重要结果是特征空间的维数不再影响计算。当不再显式表达特征向量，而是通过计算核函数的值来计算内积时所需算子数目不一定与特征的数目成比例。核的使用使得将数据隐式表达为特征空间，并在其中训练一个线性学习器成为可能，从而越过了本来需要的计算特征映射的问题。关于训练样例的惟一信息是它们在特征空间的 Gram 矩阵（见评注 2.11）。这个矩阵又称为核矩阵，本书使用符号 **K** 来表示。这个方法的关键是找到一个可以高效计算的核函数。一旦有了这个函数，决策规则可以用通过对核的 ℓ 次计算来得到：

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

使用核函数时一件有意思的事情是不需要为了在特征空间中学习而了解潜在的特征映射！本章剩余部分将考虑核函数的创建问题。考虑它们需要满足的性质，也会介绍一些创建它们的最近开发的方法。核的概念是本书写作的中心点，但不是一个通过直觉就能直接得到的想法。首先要注意的是核的思想推广了输入空间的标准内积。很明显内积本身就是一个利用单位矩阵进行特征映射的例子：

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} \cdot \mathbf{z} \rangle$$

同样也可以使用某个矩阵 **A** 通过任意固定的线性变换 $\mathbf{x} \mapsto \mathbf{Ax}$ 进行特征映射。这种情况下的核函数如下：

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{Ax} \cdot \mathbf{Az} \rangle = \mathbf{x}'\mathbf{A}'\mathbf{Az} = \mathbf{x}'\mathbf{Bz}$$

这里 $\mathbf{B} = \mathbf{A}'\mathbf{A}$ 是一个平方对称的半正定矩阵。就像在前言中介绍的一样，目的是为了将非线性引入到特征空间映射中。下面看一个简单直观的例子，这个例子通过考虑下面的关系获得非线性映射：

$$\begin{aligned} \langle \mathbf{x} \cdot \mathbf{z} \rangle^2 &= \left(\sum_{i=1}^{n} x_i z_i \right)^2 = \left(\sum_{i=1}^{n} x_i z_i \right) \left(\sum_{j=1}^{n} x_j z_j \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j z_i z_j = \sum_{(i,j)=(1,1)}^{(n,n)} \left(x_i x_j \right) \left(z_i z_j \right) \end{aligned}$$

这等价子特征向量的内积：

$$\phi(\mathbf{x}) = \left(x_i x_j \right)_{(i,j)=(1,1)}^{(n,n)}$$

在这种情况下，特征都是在例 3.3 中考虑的自由度为 2 的单项式。注意当$i \neq j$时，特征$x_i x_j$出现两次，是特征x_i^2权重的两倍。一个更一般的特征空间可以考虑这样的核：

$$\begin{aligned}\left(\langle \mathbf{x} \cdot \mathbf{z} \rangle + c\right)^2 &= \left(\sum_{i=1}^{n} x_i z_i + c\right)\left(\sum_{j=1}^{n} x_j z_j + c\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j + 2c\sum_{i=1}^{n} x_i z_i + c^2 \\ &= \sum_{(i,j)=(1,1)}^{(n,n)} \left(x_i x_j\right)\left(z_i z_j\right) + \sum_{i=1}^{n}\left(\sqrt{2c}x_i\right)\left(\sqrt{2c}z_i\right) + c^2\end{aligned}$$

这里$\binom{n+1}{2} + n + 1 = \binom{n+2}{2}$个特征都是自由度最大为 2 的单项式，但是通过参数 c 控制的自由度为 1 或 2 的相关权重，也可以决定自由度为 0 或常数的特征。与此类似，核函数可有以下形式：

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} \cdot \mathbf{z} \rangle^d \text{和} K(\mathbf{x}, \mathbf{z}) = \left(\langle \mathbf{x} \cdot \mathbf{z} \rangle + c\right)^d$$

其中$d \geqslant 2$。对第一个核，$\binom{n+d-1}{d}$不同的特征都是自由度为 d 的单项式，尽管权重随指数的不同而变化。对第二个核，有$\binom{n+d}{d}$不同的特征，这些单项式的最大自由度为 d。输入空间的决策边界对应着这些特征空间的超平面，它们是自由度为 d 的多项式曲线，因此这些核通常称为多项式核。

更复杂的核是可能的，下节将考虑构造核的重要问题。这个问题的一个重要方面是构造核的函数$K(\mathbf{x}, \mathbf{z})$的数学特征。泛函分析的定理将为这个问题提供一个答案。

3.3　构造核函数

使用核函数是诱人的计算捷径。若希望使用这种方法，看起来似乎首先需要创立一个复杂的特征空间，然后计算出空间的内积，最后寻找一种直接的方法用原始输入计算该值。而实际上是直接定义一个核函数，通过它隐式地定义特征空间。利用这种方式，不仅在计算内积时，而且在学习器的设计中都可以避开特征空间。要强调的是为输入空间定义一个核函数通常比创立一个复杂的特征空间更自然。在实现这个思路之前，首先要决定函数$K(\mathbf{x}, \mathbf{z})$的哪些性质对于确定它是否适合某个特征空间是必要的。明显的是函数必须是对称的：

$$K(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{z}) \rangle = \langle \boldsymbol{\phi}(\mathbf{z}) \cdot \boldsymbol{\phi}(\mathbf{x}) \rangle = K(\mathbf{z}, \mathbf{x})$$

并且满足下面的不等式，这个不等式是从 Cauchy-Schwarz 不等式得到的：

$$\begin{aligned} K(\mathbf{x},\mathbf{z})^2 &= \langle \boldsymbol{\phi}(\mathbf{x})\cdot\boldsymbol{\phi}(\mathbf{z})\rangle^2 \leqslant \|\boldsymbol{\phi}(\mathbf{x})\|^2\|\boldsymbol{\phi}(\mathbf{z})\|^2 \\ &= \langle \boldsymbol{\phi}(\mathbf{x})\cdot\boldsymbol{\phi}(\mathbf{x})\rangle\langle \boldsymbol{\phi}(\mathbf{z})\cdot\boldsymbol{\phi}(\mathbf{z})\rangle = K(\mathbf{x},\mathbf{x})K(\mathbf{z},\mathbf{z}) \end{aligned}$$

然而，这些条件对于保证特征空间的存在是不充分的。

3.3.1 核函数的性质

Mercer 定理

本节将介绍 Mercer 定理，它刻画了函数$K(\mathbf{x},\mathbf{z})$是核函数时的性质。我们从考虑一个简单的实例开始，引导得到最后的结果。首先考虑有限输入空间$X=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$，并假定$K(\mathbf{x},\mathbf{z})$是在$X$上的对称函数。考虑矩阵：

$$\mathbf{K}=\left(K(\mathbf{x}_i,\mathbf{x}_j)\right)_{i,j=1}^n$$

既然$\mathbf{K}$是对称的，必存在一个正交矩阵$\mathbf{V}$使得$\mathbf{K}=\mathbf{V}\Lambda\mathbf{V}'$，这里$\Lambda$是包含$\mathbf{K}$的特征值$\lambda_t$的对角矩阵，特征值$\lambda_t$对应着特征向量$\mathbf{v}_t=(v_{ti})_{i=1}^n$，也就是$\mathbf{V}$的列。现在假定所有特征值是非负的，考虑特征映射：

$$\boldsymbol{\phi}:\mathbf{x}_i\mapsto\left(\sqrt{\lambda_t}v_{ti}\right)_{t=1}^n\in\mathbb{R}^n \quad i=1,\ldots,n$$

现在有：

$$\langle\boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j)\rangle=\sum_{t=1}^n\lambda_t v_{ti}v_{tj}=\left(\mathbf{V}\Lambda\mathbf{V}'\right)_{ij}=\mathbf{K}_{ij}=K(\mathbf{x}_i,\mathbf{x}_j)$$

这意味着$K(\mathbf{x},\mathbf{z})$是真正对应于特征映射$\boldsymbol{\phi}$的核函数。$\mathbf{K}$的特征值非负的条件是必要的，因为如果有一个负特征值λ_s对应着特征向量$\mathbf{v}_s$，特征空间中的点：

$$\mathbf{z}=\sum_{i=1}^n v_{si}\boldsymbol{\phi}(\mathbf{x}_i)=\sqrt{\Lambda}\mathbf{V}'\mathbf{v}_s$$

有二阶范数：

$$\|\mathbf{z}\|^2=\langle\mathbf{z}\cdot\mathbf{z}\rangle=\mathbf{v}_s'\mathbf{V}\sqrt{\Lambda}\sqrt{\Lambda}\mathbf{V}'\mathbf{v}_s=\mathbf{v}_s'\mathbf{V}\Lambda\mathbf{V}'\mathbf{v}_s=\mathbf{v}_s'\mathbf{K}\mathbf{v}_s=\lambda_s<0$$

与空间的几何性质相矛盾。因此得出下面的命题。

命题 3.5 令X是有限输入空间，$K(\mathbf{x},\mathbf{z})$是X上的对称函数。那么$K(\mathbf{x},\mathbf{z})$是核函数的充分必要条件是矩阵

$$\mathbf{K}=\left(K(\mathbf{x}_i,\mathbf{x}_j)\right)_{i,j=1}^n$$

是半正定的（即特征值非负）。

从这个简单的例子得到启发，可以在希尔伯特（Hilbert）空间通过为每个特征引入权重λ_i推广内积，得到：

$$\langle \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{z}) \rangle = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{z}) = K(\mathbf{x}, \mathbf{z})$$

这样特征向量变为：

$$\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_n(\mathbf{x}), \ldots)$$

Mercer 定理给出了连续对称函数$K(\mathbf{x}, \mathbf{z})$允许的表示方式：

$$K(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{z})$$

的充分必要条件，其中λ_i非负，它等价于$K(\mathbf{x}, \mathbf{z})$是特征空间$F \supseteq \boldsymbol{\phi}(X)$中的内积，这里$F$是下面所有序列的$l_2$空间：

$$\boldsymbol{\psi} = (\psi_1, \psi_2, \ldots, \psi_i, \ldots)$$

其中：

$$\sum_{i=1}^{\infty} \lambda_i \psi_i^2 < \infty$$

它将隐式地得出一个特征向量定义的空间，作为结果，像第 2 章中描述的那些F中的线性函数，将这样表示：

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i \psi_i \phi_i(\mathbf{x}) + b = \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}, \mathbf{x}_j) + b$$

这里第一个表达式是函数原空间中的表示，第二个是对偶空间中的表示，两者的关系是：

$$\boldsymbol{\psi} = \sum_{j=1}^{\ell} \alpha_j y_j \boldsymbol{\phi}(\mathbf{x}_j)$$

注意在原空间中的表达式中加和项的数目等于特征空间的维数，而对偶空间中项的数目是ℓ个（等于样本数目）。从特征空间的大小来考虑，两个之中的一个表示可能比另一个更便利。明显的是，只要核函数是非线性的，从形式上看这个函数就是非线性的。

在有限输入空间情况下与之类似。泛函分析的贡献在于研究如下形式的积分方程的特征值问题：

$$\int_X K(\mathbf{x},\mathbf{z})\phi(\mathbf{z})d\mathbf{z} = \lambda\phi(\mathbf{x})$$

这里$K(\mathbf{x},\mathbf{z})$是有界、对称、正定核函数，X是一个紧凑空间。本书不再深入到分析的细节，只简单地给出定理（符号和定义见附录B.3）。

定理 3.6 （Mercer）令 X 是$\mathbb{R}^n$的紧子集。假定 K 是连续对称函数，存在积分算子 $T_K : L_2(X) \to L_2(X)$，使得：

$$(T_K f)(\cdot) = \int_X K(\cdot,\mathbf{x})f(\mathbf{x})d\mathbf{x}$$

是正的，也就是：

$$\int_{X\times X} K(\mathbf{x},\mathbf{z})f(\mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} \geqslant 0 \quad \forall f \in L_2(X)$$

对所有的$f \in L_2(X)$成立。然后扩展$K(\mathbf{x},\mathbf{z})$到一个一致收敛的序列（在$X \times X$上），这个序列由T_K的特征函数$\phi_j \in L_2(X)$构成，归一化使得$\|\phi_j\|_{L_2} = 1$，并且$\lambda_j \geqslant 0$，则有：

$$K(\mathbf{x},\mathbf{z}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x})\phi_j(\mathbf{z})$$

评注 3.7 正条件：

$$\int_{X\times X} K(\mathbf{x},\mathbf{z})f(\mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} \geqslant 0$$

对应着有限输入空间情况下的半正定条件。在点集$\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$上的有限矩阵条件可以通过将$f$选择为$\mathbf{x}_i$的$\delta$函数的加权和来重新形成。既然这样的函数是$L_2(X)$中函数的上下限，上面的条件意味着对任意 X 的有限子集，相应的矩阵是半正定的。逆命题也是真的，因为如果正条件不在某个函数f上成立，则可以用输入的有限和来逼近积分，如果选择得十分恰当，就会给出一个负值。比如在所选网点$\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$上的$f$值形成一个向量$\mathbf{v}$，相应的核矩阵$\mathbf{K}$满足：

$$\mathbf{v}'\mathbf{K}\mathbf{v} < 0$$

显示$\mathbf{K}$不是半正定的。Mercer定理的条件等价于对X的任意有限子集，相应的矩阵是半正定的命题。这就是核函数的第二个特征，这个特征将来在构造核时最有用。核一般是指满足这个性质的函数，文献中通常称为Mercer核。

评注 3.8 沿着命题3.5之后的内积定义展开，定理建议映射特征：

$$\mathbf{x} = (x_1,\ldots,x_n) \longmapsto \phi(\mathbf{x}) = (\phi_1(\mathbf{x}),\ldots,\phi_j(\mathbf{x}),\ldots)$$

到下面的加权内积定义的希尔伯特空间：

$$\left\langle \psi \cdot \tilde{\psi} \right\rangle = \sum_{j=1}^{\infty} \lambda_j \psi_j \tilde{\psi}_j$$

既然是两个特征向量的内积，则满足：

$$\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{z}) = K(\mathbf{x}, \mathbf{z})$$

这些特征的特殊性质在于它们都是 $L_2(X)$中的标准正交函数。这种映射有时称为 Mercer 特征。为了确保谱是可数集，输入域应是紧的，注意并不要求特征形成正交集。上面给出的有限输入空间的例子中，特征已经被特征值的平方根重新尺度化。一般可以重新尺度化每个坐标：

$$\mathbf{x} = (x_1, \ldots, x_n) \longmapsto \phi(\mathbf{x}) = (b_1 \phi_1(\mathbf{x}), \ldots, b_j \phi_j(\mathbf{x}), \ldots)$$

到下面的加权内积定义的希尔伯特空间：

$$\left\langle \psi \cdot \tilde{\psi} \right\rangle = \sum_{j=1}^{\infty} \frac{\lambda_j}{b_j^2} \psi_j \tilde{\psi}_j$$

既然又是两个特征向量的内积，则满足：

$$\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle = \sum_{j=1}^{\infty} \frac{\lambda_j}{b_j^2} b_j \phi_j(\mathbf{x}) b_j \phi_j(\mathbf{z}) = K(\mathbf{x}, \mathbf{z})$$

在这种情况下，有限输入空间例子中的特征仍然是正交的。然而，正交性是不需要的。比如在多项式核的情况下，例 3.3 给出的特征一般不正交。特征可以选择正交，对多项式核需要选择适当的$\mathbb{R}^2$的子集作为输入域并选择适当的积分测度。

Mercer 特征提供了一个输入点的表示方式，它通过核的有限数目的特征值定义的内积特征空间的映像来实现。输入空间映像形成的子流形由核算于的特征向量定义。点$\phi(\mathbf{x}) \in F$的映像不需要计算，因为它们的内积可以使用核函数计算。

例 3.9　考虑核函数 $K(\mathbf{x}, \mathbf{z}) = K(\mathbf{x} - \mathbf{z})$。若两者是平移相同的向量，两个输入的内积是不变的，所以说这样的核具有平移不变性。考虑一维情况下，K 定义在$[0, 2\pi]$上，$K(u)$可以在$\mathbb{R}$上扩展成连续、对称、周期性的函数。这样的函数可以展成一致收敛的傅里叶序列：

$$K(u) = \sum_{n=0}^{\infty} a_n \cos(nu)$$

在这种情况下，可将$K(x-z)$展开为：

$$K(x-z)=a_0+\sum_{n=1}^{\infty}a_n\sin(nx)\sin(nz)+\sum_{n=1}^{\infty}a_n\cos(nx)\cos(nz)$$

假定a_n全是正的，这显示$K(x,z)$是作为下面正交特征定义的特征空间中的内积：

$$\{\phi_i(x)\}_{i=0}^{\infty}=(1,\sin(x),\cos(x),\sin(2x),\cos(2x),\ldots,\sin(nx),\cos(nx),\ldots)$$

因为这些函数$\cos(nu)$和$\sin(nu)$形成了一套在区间$[0,2\pi]$上的正交函数集合。因此，将其归一化将得到一套 Mercer 特征。注意内嵌的特征定义与a_n无关，因此a_n将控制特征空间的几何性质。

从例 3.9 可以看到选择核的重要性。$K(u)$展开式中的参数a_n是傅里叶系数。如果对某个n，有$a_n=0$，相应的特征可以从特征空间去除。类似地，a_n中小的值意味着特征具有小的权重，在超平面的选择中具有小的影响。因此，核的选择可以视做选择特定谱特性的滤波器，它的作用是控制不同频率在优化分类面时的影响。下个小节将从其在编码学习偏置的角度介绍特征空间。

给定一个特征空间的表示方式，它有多个（可数）线性无关特征（不一定是正交的或单位化的），这些特征是映射：

$$\mathbf{x}=(x_1,\ldots,x_n)\longmapsto\boldsymbol{\phi}(\mathbf{x})=(\phi_1(\mathbf{x}),\ldots,\phi_j(\mathbf{x}),\ldots)$$

到下面加权内积定义的l_2空间F中得到的：

$$\left\langle\psi\cdot\tilde{\psi}\right\rangle=\sum_{j=1}^{\infty}\mu_j\psi_j\tilde{\psi}_j$$

在输入空间中定义函数$\mathscr{H}$的空间，使其成为映射：

$$T:\psi\longmapsto\sum_{j=1}^{\infty}\psi_j\phi_j(\mathbf{x}) \tag{3.1}$$

下F的映像。注意，如果F是有限维，$\mathscr{H}$是在输入空间上的函数类。既然它是基函数的所有线性组合的集合，则可以在特征空间中通过应用线性函数有效使用。在无限维特征空间，$\mathscr{H}$可能不包含所有可能的假设函数，因为它们可能是在F中没有有限范数的点的映像，或者等价地说$\mathscr{H}$也可能有过多的函数。下个小节将考虑特征映射的某种选择，它能保证$\mathscr{H}$确切地包含假设集，同时有一定数目的附加性质。

再生核希尔伯特空间

假定特征空间是通过映射：

$$\mathbf{x}=(x_1,\ldots,x_n)\longmapsto\boldsymbol{\phi}(\mathbf{x})=(\phi_1(\mathbf{x}),\ldots,\phi_j(\mathbf{x}),\ldots)$$

到下面加权内积定义的l_2空间 F 中得到：

$$\left\langle\boldsymbol{\psi}\cdot\tilde{\boldsymbol{\psi}}\right\rangle=\sum_{j=1}^{\infty}\mu_j\psi_j\tilde{\psi}_j$$

这里：

$$K(\mathbf{x},\mathbf{z})=\sum_{i=1}^{\infty}\mu_i\phi_i(\mathbf{x})\phi_i(\mathbf{z})$$

现在考虑引入同比于因子μ_i的一个权重，这样映像有下面的形式：

$$\mathbf{x}=(x_1,\ldots,x_n)\longmapsto\boldsymbol{\psi}(\mathbf{x})=(\psi_1(\mathbf{x}),\ldots,\psi_j(\mathbf{x}),\ldots)=(\mu_1\phi_1(\mathbf{x}),\ldots,\mu_j\phi_j(\mathbf{x}),\ldots)$$

选择适当的加权内积如下式给出：

$$\left\langle\boldsymbol{\psi}\cdot\tilde{\boldsymbol{\psi}}\right\rangle_{\mu}=\sum_{j=1}^{\infty}\frac{\psi_j\tilde{\psi}_j}{\mu_j}$$

这样：

$$\|\boldsymbol{\psi}(\mathbf{x})\|_{\mu}^2=K(\mathbf{x},\mathbf{x})$$

用$\mathcal{H}$表示由式(3.1)定义的映射 T 下的 F 的映像。这个特定的加权有若干特殊性质。对两个函数：

$$f(\mathbf{x})=\sum_{j=1}^{\infty}\psi_j\phi_j(\mathbf{x})\text{和}g(\mathbf{x})=\sum_{j=1}^{\infty}\tilde{\psi}_j\phi_j(\mathbf{x})$$

在$\mathcal{H}$中定义内积：

$$\langle f(\cdot)\cdot g(\cdot)\rangle_{\mathcal{H}}=\sum_{j=1}^{\infty}\frac{\psi_j\tilde{\psi}_j}{\mu_j}$$

因此使得映射 T 等距。可以看到如果将输入点映射到特征空间，然后应用 T 到这个映像，可以得到：

$$T(\boldsymbol{\psi}(\mathbf{z}))=\sum_{j=1}^{\infty}\psi_j(\mathbf{z})\phi_j(\mathbf{x})=\sum_{j=1}^{\infty}\mu_j\phi_j(\mathbf{z})\phi_j(\mathbf{x})=K(\mathbf{z},\mathbf{x})$$

这样$K(\mathbf{z},\cdot)\in\mathcal{H}$。因此用对偶表示的函数为：

$$f(\mathbf{x})=\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\mathbf{x})$$

是在$\mathscr{H}$中的。进一步，如果取通用函数$f(\mathbf{x})=\sum_{j=1}^{\infty}a_j\phi_j(\mathbf{x})\in\mathscr{H}$与函数$K(\mathbf{x},\mathbf{z})$的内积，可以得到：

$$\begin{aligned}\langle f(\cdot)\cdot K(\mathbf{z},\cdot)\rangle_{\mathscr{H}} &= \left\langle (a_j)_{j=1}^{\infty}\cdot(\mu_j\phi_j(\mathbf{z}))_{j=1}^{\infty}\right\rangle_{\mu}\\ &= \sum_{j=1}^{\infty}\frac{a_j\mu_j\phi_j(\mathbf{z})}{\mu_j}=\sum_{j=1}^{\infty}a_j\phi_j(\mathbf{z})=f(\mathbf{z})\end{aligned}$$

这就是所知的核K的再生性。这也意味着$\mathscr{H}$同下面子空间的闭包是一致的：

$$\mathscr{X}=\left\{\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\mathbf{x}):\ell\in\mathbb{N},\ (\mathbf{x}_1,\ldots,\mathbf{x}_\ell)\in X^\ell,\alpha_i\in\mathbb{R}\right\}$$

如果$f\in\mathscr{H}$并且满足$f\perp\mathscr{X}$，则对所有的$\mathbf{z}\in X$有：

$$f(\mathbf{z})=\langle f(\cdot)\cdot K(\mathbf{z},\cdot)\rangle_{\mathscr{H}}=0$$

这意味着$f=0$。因此$\mathscr{H}$包含在$\mathscr{X}$的闭包中，并且不会包含对偶表示中不能任意逼近的函数。对于对偶空间中的两个函数$f(\mathbf{x})=\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\mathbf{x})$和$g(\mathbf{x})=\sum_{j=1}^{\hat{\ell}}\hat{\alpha}_jK(\hat{\mathbf{x}}_j,\mathbf{x})$。内积由下式给出：

$$\begin{aligned}\langle f(\cdot)\cdot g(\cdot)\rangle_{\mathscr{H}} &= \left\langle\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\cdot)\cdot\sum_{j=1}^{\hat{\ell}}\hat{\alpha}_jK(\hat{\mathbf{x}}_j,\cdot)\right\rangle_{\mathscr{H}}\\ &= \sum_{i=1}^{\ell}\alpha_i\sum_{j=1}^{\hat{\ell}}\hat{\alpha}_j\langle K(\mathbf{x}_i,\cdot)\cdot K(\hat{\mathbf{x}}_j,\cdot)\rangle_{\mathscr{H}}\\ &= \sum_{i=1}^{\ell}\alpha_i\sum_{j=1}^{\hat{\ell}}\hat{\alpha}_jK(\mathbf{x}_i,\hat{\mathbf{x}}_j)\\ &= \sum_{i=1}^{\ell}\alpha_ig(\mathbf{x}_i)=\sum_{j=1}^{\hat{\ell}}\hat{\alpha}_jf(\hat{\mathbf{x}}_j)\end{aligned}\tag{3.2}$$

这显示内积的定义与函数特定的表示形式无关[改变g的表示形式不是改变$g(\mathbf{x}_i)$的值]。另外，可以得到$\|f\|_{\mathscr{H}}^2=\sum_{i=1}^{\ell}\alpha_if(\mathbf{x}_i)$，可以看到要得到$f$的有界范数，一定要有界的值和系数。最后，要注意的是再生性质$\langle f(\cdot)\cdot K(\mathbf{z},\cdot)\rangle_{\mathscr{H}}=f(\mathbf{z})$意味着由$F_{\mathbf{x}}[f]=f(\mathbf{x})$ $(\forall f\in\mathscr{H})$定义的评价泛函是线性有界的，存在$U_{\mathbf{z}}=\|K(\mathbf{z},\cdot)\|_{\mathscr{H}}\in\mathbb{R}^+$，这样由Cauchy-Schwarz不等式可以得到：

$$|F_{\mathbf{z}}[f]|=|f(\mathbf{z})|=\langle f(\cdot)\cdot K(\mathbf{z},\cdot)\rangle_{\mathscr{H}}\leqslant U_{\mathbf{z}}\|f\|_{\mathscr{H}}$$

对所有$f\in\mathscr{H}$成立。

对定义在输入域 $X \subset \mathbb{R}^d$ 上的函数的希尔伯特空间 $\mathscr{H}$，评价泛函的线性有界是再生核希尔伯特空间（RKHS, reproducing kernel Hilbert space）定义的性质。因此可以得出下面的结论。

定理 3.10　*对定义在域 $X \subset \mathbb{R}^d$ 上的每一个 Mercer 核 $K(\mathbf{x},\mathbf{z})$，存在一个定义在 X 上的函数的 RKHS $\mathscr{H}$，其中 K 是再生核。*

该定理的逆定理也成立。也就是对线性有界函数的任意希尔伯特空间，存在再生核函数。再生核也是 Mercer 核，这可以由下面事实得出，对于 $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x})$：

$$\begin{aligned} 0 \leqslant \|f\|_{\mathscr{H}}^2 &= \left\langle \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \cdot) \cdot \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \cdot) \right\rangle_{\mathscr{H}} \\ &= \sum_{i=1}^{\ell} \alpha_i \sum_{j=1}^{\ell} \alpha_j \left\langle K(\mathbf{x}_i, \cdot) \cdot K(\mathbf{x}_j, \cdot) \right\rangle_{\mathscr{H}} \\ &= \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

意味着对每个 X 的有限子集，K 是半正定的，并且可由评注 3.7 充分得出算子 K 的正性。

下面的例子显示了其中的一些力量并探究 RKHS 构造函数所能做的。

例 3.11　假定在下面的训练点集基础上做回归：

$$S = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\ell, y_\ell)) \subset (X \times Y)^\ell \subset (\mathbb{R}^n \times \mathbb{R})^\ell$$

这个数据集由目标函数 $t(\mathbf{x})$ 产生。如果假定一个对偶表示形式：

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

可以寻找最小化范数：

$$\begin{aligned} \|f - t\|_{\mathscr{H}}^2 &= \left\langle \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) - t(\mathbf{x}) \cdot \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) - t(\mathbf{x}) \right\rangle_{\mathscr{H}} \\ &= -2 \left\langle t(\mathbf{x}) \cdot \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \right\rangle_{\mathscr{H}} + \|f\|_{\mathscr{H}}^2 + \|t\|_{\mathscr{H}}^2 \\ &= -2 \sum_{i=1}^{\ell} \alpha_i \left\langle t(\mathbf{x}) \cdot K(\mathbf{x}_i, \mathbf{x}) \right\rangle_{\mathscr{H}} + \|f\|_{\mathscr{H}}^2 + \|t\|_{\mathscr{H}}^2 \\ &= -2 \sum_{i=1}^{\ell} \alpha_i y_i + \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + \|t\|_{\mathscr{H}}^2 \end{aligned}$$

既然$\|t\|_{\mathcal{H}}^2$与它们有关，这可以通过令对应参数的导数为 0 来求解。本书将在第 5 章例 5.6 深入讨论。注意如果将 K 视做 Mercer 核，特征空间的 Gram 矩阵将如何出现。

3.3.2 从核函数中构造核函数

确认一个新的对称函数是一个核函数的关键是评注 3.7 列出的条件，它要求函数在任意有限点集上定义的矩阵是半正定的。应用这个条件来确认几个新的核是不是 Mercer 核。从下面的命题可看出核满足一定数目的闭性质，它允许从简单的构件块创立复杂的核。

命题 3.12 令K_1和K_2是在$X \times X$上的核，$X \subseteq \mathbb{R}^n$，$a \in \mathbb{R}^+$，$f(\cdot)$是X上的一个实值函数：

$$\phi : X \longrightarrow \mathbb{R}^m$$

K_3是$\mathbb{R}^m \times \mathbb{R}^m$上的核，并且$\mathbf{B}$是一个对称半正定$n \times n$矩阵。那么下面的函数是核函数：

1. $K(\mathbf{x},\mathbf{z}) = K_1(\mathbf{x},\mathbf{z}) + K_2(\mathbf{x},\mathbf{z})$
2. $K(\mathbf{x},\mathbf{z}) = aK_1(\mathbf{x},\mathbf{z})$
3. $K(\mathbf{x},\mathbf{z}) = K_1(\mathbf{x},\mathbf{z})K_2(\mathbf{x},\mathbf{z})$
4. $K(\mathbf{x},\mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$
5. $K(\mathbf{x},\mathbf{z}) = K_3(\phi(\mathbf{x}),\phi(\mathbf{z}))$
6. $K(\mathbf{x},\mathbf{z}) = \mathbf{x}'\mathbf{B}\mathbf{z}$

证明 固定一个有限点集$\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$，令$\mathbf{K}_1$和$\mathbf{K}_2$是限制在这些点上的相应的矩阵。考虑任意向量$\alpha \in \mathbb{R}^\ell$。回顾矩阵$\mathbf{K}$是半正定矩阵的充分必要条件是：对所有$\alpha$，有$\alpha'\mathbf{K}\alpha \geqslant 0$。

1. 有：

$$\alpha'(\mathbf{K}_1 + \mathbf{K}_2)\alpha = \alpha'\mathbf{K}_1\alpha + \alpha'\mathbf{K}_2\alpha \geqslant 0$$

这样$\mathbf{K}_1 + \mathbf{K}_2$是半正定的，而$K_1 + K_2$是核函数。

2. 类似地，$\alpha' a\mathbf{K}_1\alpha = a\alpha'\mathbf{K}_1\alpha \geqslant 0$，$aK_1$是核。

3. 令：

$$\mathbf{K} = \mathbf{K}_1 \otimes \mathbf{K}_2$$

是矩阵$\mathbf{K}_1$和$\mathbf{K}_2$的张量积，两个半正定矩阵的张量积是半正定的，因为积的特征值是两个成分的特征值的成对的积。函数K_1K_2对应的矩阵就是$\mathbf{K}_1$和$\mathbf{K}_2$的所知的 Schur 积$\mathbf{H}$，它的项是两个成分的对应项的积。矩阵$\mathbf{H}$是由一个列集

合和相同集合的行定义的 $\mathbf{K}$ 的主子矩阵。因此对任意 $\boldsymbol{\alpha} \in \mathbb{R}^{\ell}$，有相应的 $\boldsymbol{\alpha}_1 \in \mathbb{R}^{\ell^2}$，满足

$$\boldsymbol{\alpha}'\mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\alpha}_1'\mathbf{K}\boldsymbol{\alpha}_1 \geqslant 0$$

这样 $\mathbf{H}$ 就是所需的半正定矩阵。

4. 如下重新排列双线性形式：

$$\begin{aligned}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j) &= \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j f(\mathbf{x}_i)f(\mathbf{x}_j)\\ &= \sum_{i=1}^{\ell}\alpha_i f(\mathbf{x}_i)\sum_{j=1}^{\ell}\alpha_j f(\mathbf{x}_j)\\ &= \left(\sum_{i=1}^{\ell}\alpha_i f(\mathbf{x}_i)\right)^2 \geqslant 0\end{aligned}$$

5. 既然 K_3 是核函数，限制 K_3 到点集 $\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_\ell)$ 上得到的矩阵是半正定的。
6. 考虑使用正交矩阵 $\mathbf{V}$ 对角化 $\mathbf{B} = \mathbf{V}'\Lambda\mathbf{V}$，这里 $\mathbf{A}$ 是包含非负特征值的对角矩阵。令 $\sqrt{\Lambda}$ 是有特征值平方根的对角矩阵，设置 $\mathbf{A} = \sqrt{\Lambda}\mathbf{V}$。因而得到：

$$K(\mathbf{x},\mathbf{z}) = \mathbf{x}'\mathbf{B}\mathbf{z} = \mathbf{x}'\mathbf{V}'\Lambda\mathbf{V}\mathbf{z} = \mathbf{x}'\mathbf{V}'\sqrt{\Lambda}\sqrt{\Lambda}\mathbf{V}\mathbf{z} = \mathbf{x}'\mathbf{A}'\mathbf{A}\mathbf{z} = \langle \mathbf{A}\mathbf{x}\cdot\mathbf{A}\mathbf{z}\rangle$$

这是使用特征映射 $\mathbf{A}$ 的内积。

推论 3.13　令 $K_1(\mathbf{x},\mathbf{z})$ 是在 $X \times X$ 上的核，$\mathbf{x}, \mathbf{z} \in X$，$p(x)$ 是正系数的多项式。则下面的函数同样是核：

1. $K(\mathbf{x},\mathbf{z}) = p(K_1(\mathbf{x},\mathbf{z}))$
2. $K(\mathbf{x},\mathbf{z}) = \exp(K_1(\mathbf{x},\mathbf{z}))$
3. $K(\mathbf{x},\mathbf{z}) = \exp(-\|\mathbf{x}-\mathbf{z}\|^2/\sigma^2)$

证明　依次证明三个部分：

1. 对于多项式的结论，可以通过组合命题的部分结论直接得到。注意，常数可以使用命题的第四部分。
2. 幂函数可以用正系数的多项式任意逼近，因此是核的一个极限。而点对限制下核明显是闭包，得到结论。
3. 可分解高斯函数如下：

$$\exp(-\|\mathbf{x}-\mathbf{z}\|^2/\sigma^2) = \exp(-\|\mathbf{x}\|^2/\sigma^2)\exp(-\|\mathbf{z}\|^2/\sigma^2)\exp(2\langle\mathbf{x}\cdot\mathbf{z}\rangle/\sigma^2)$$

可从命题的第四部分得出前两个因子是核，可从本推论的第二个部分得出第三个因子是核。

评注 3.14 推论中最后讨论的核是高斯核。这个函数形成了径向基函数网络的核心，因此使用这个核意味着假设是径向基函数网络。

3.3.3 从特征中构造核函数

另一个得到核的方式当然是从特征开始，通过计算内积得到。这种情况下，不需要检查半正定性，因为这是自动从内积作为定义开始的。前面给出的第一个关于多项式核的例子就是沿着这个思路。现在提出一个定义在离散空间的不寻常的例子，即有限字符串核，这是为了例示非欧氏空间中该方法的潜力。

例 3.15 （字符串子序列核）令Σ是一个有限字符表。字符串是从Σ中取出的有限个字符的序列，包含空序列。对字符串 s，t，用$|s|$代表字符串$s = s_1 \ldots s_{|s|}$的长度，st代表字符串s和t的连接。字符串$s[i:j]$是s的子串$s_i \ldots s_j$。如果存在下标$\mathbf{i} = (i_1, \ldots, i_{|u|})$，并且$1 \leqslant i_1 < \cdots < i_{|u|} \leqslant |s|$，对$j = 1, \ldots, |u|$满足$u_j = s_{i_j}$或为了简洁$u = s[\mathbf{i}]$，则称$u$是$s$的子序列。$s$中序列的长度$l(\mathbf{i})$是$i_{|u|} - i_1 + 1$。用$\Sigma^n$代表长度$n$的有限字符串，$\Sigma^*$代表所有字符串的集合：

$$\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$$

现在定义特征空间$F_n = \mathbb{R}^{\Sigma^n}$。可以通过为每个$u \in \Sigma^n$定义$u$坐标$\phi_u(s)$给出字符串$s$的特征映射$\boldsymbol{\phi}$。对某个$\lambda \leqslant 1$，定义：

$$\phi_u(s) = \sum_{\mathbf{i}:u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})}$$

这些特征度量了字符串s中子字符串的出现次数，并根据它们的长度加权。因此两个字符串s, t特征向量的内积给出了所有子串根据它们的频率和长度加权的总和：

$$\begin{aligned} K_n(s,t) &= \sum_{u \in \Sigma^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \\ &= \sum_{u \in \Sigma^n} \sum_{\mathbf{i}:u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})} \sum_{\mathbf{j}:u=t[\mathbf{j}]} \lambda^{l(\mathbf{j})} \\ &= \sum_{u \in \Sigma^n} \sum_{\mathbf{i}:u=s[\mathbf{i}]} \sum_{\mathbf{j}:u=t[\mathbf{j}]} \lambda^{l(\mathbf{i})+l(\mathbf{j})} \end{aligned}$$

这样的核明显对于文本分类很有用，但是一眼就可以看出实际计算量很大。下面介绍一个附加函数来辅助定义一个回归核。令：

$$K'_i(s,t)=\sum_{u\in\Sigma^i}\sum_{i:u=s[\mathbf{i}]}\sum_{j:u=t[\mathbf{j}]}\lambda^{|s|+|t|-i_1-j_1+2}\quad i=1,\ldots,n-1$$

它计数了到字符串 s 和 t 结束的长度，而不仅仅是$l(\mathbf{i})$和$l(\mathbf{j})$。现在定义一个K'_i的回归计算，用来计算K_n：

$$\begin{aligned}
K'_0(s,t)&=1,\quad \text{对所有 } s,t\\
K'_i(s,t)&=0,\quad \text{如果 } \min\left(|s|,|t|\right)<i\\
K_i(s,t)&=0,\quad \text{如果 } \min\left(|s|,|t|\right)<i\\
K'_i(sx,t)&=\lambda K'_i(s,t)+\sum_{j:t_j=x}K'_{i-1}(s,t[1:j-1])\lambda^{|t|-j+2}\quad i=1,\ldots,n-1\\
K_n(sx,t)&=K_n(s,t)+\sum_{j:t_j=x}K'_{n-1}(s,t[1:j-1])\lambda^2
\end{aligned}$$

这个回归的校正是从观察字符串长度如何增长得来的，它为每个额外的字符增加了一个λ因子，直到获得 n 个字符的全部长度。令人惊奇的是回归方程以跟$n|s||t|$成比例的时间计算核，时间是足够的。如果想计算不同 n 值的$K_n(s,t)$，仅仅将$K'_i(s,t)$的计算增加到所要求的最大 n 值减 1 即可。当然可以使用命题 3.12 的部分 1 和部分 2 创立一个核$K(s,t)$，使得对不同的 n 可以给出不同的权重来组合不同的$K_n(s,t)$。

3.4　特征空间中的计算

既然特征映射：

$$\mathbf{x}=(x_1,\ldots,x_n)\longmapsto\boldsymbol{\phi}(\mathbf{x})=(\phi_1(\mathbf{x}),\ldots,\phi_j(\mathbf{x}),\ldots)$$

得到的内嵌是非线性的，一般来说它定义了特征空间的 n 维子流形，映像的线性组合通常不对应任意输入点的映像。不用说，这些点也可以使用对偶表示，如果只关心距离和内积，可以不需要显式计算特征向量就可以使用核。本节简要回顾了计算如何进行。令$\boldsymbol{\phi}(X)$是在特征映射下输入空间的映像，并定义$F=\mathrm{co}(\boldsymbol{\phi}(X))$为$\boldsymbol{\phi}(X)$中点的所有有限线性组合的空间。在 F 中，一个广义的点可以表示为：

$$P=\sum_{i=1}^{\ell}\alpha_i\boldsymbol{\phi}(\mathbf{x}_i)$$

这是通过序列对表示的：

$$P=(\alpha_i,\mathbf{x}_i)_{i=1}^{\ell}$$

考虑第二个这样的点 $Q=(\beta_i,\mathbf{z}_i)_{i=1}^{s}$。两个这样点的和与差是通过点集的对应的系数

加或减得到。比如，如果 $\mathbf{x}_i$, $i=1,\ldots,\ell$ 是一个正样本的集合，$\mathbf{z}_i$, $i=1,\ldots,s$ 是负样本的集合，则从负样本的质心到正样本的质心所确定的超平面的权重向量是：

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(1,\mathbf{x}_i)-\frac{1}{s}\sum_{j=1}^{s}(1,\mathbf{z}_j)=\left(\frac{1}{\ell},\mathbf{x}_i\right)_{i=1}^{\ell}\cup\left(-\frac{1}{s},\mathbf{z}_j\right)_{j=1}^{s}$$

与第二个点 $Q=(\beta_i,\mathbf{z}_i)_{i=1}^{s}$ 的内积由下式给出：

$$\langle P\cdot Q\rangle_F=\sum_{i,j}\alpha_i\beta_jK(\mathbf{x}_i,\mathbf{z}_j)$$

注意这刚好对应着相应 RKHS $\mathscr{H}$ 中的两个函数：

$$f(\mathbf{x})=\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\mathbf{x})\text{ 和 }g(\mathbf{x})=\sum_{i=1}^{s}\beta_iK(\mathbf{z}_i,\mathbf{x})$$

的内积 $\langle f\cdot g\rangle_{\mathscr{H}}$。因此，也可以将对的序列视为 RKHS 空间的表示函数，其内积给出了相应的 RKHS 内积。

对点 $\mathbf{x}\in X$，特征向量的范数平方是 $K(\mathbf{x},\mathbf{x})$。注意对于高斯核，在所有输入上它等于 1。在这种情况下，表示输入空间嵌入到单位球的表面。类似地，P 和 Q 之间距离的平方可以计算如下：

$$\begin{aligned}\|P-Q\|_F^2 &= \langle P-Q\cdot P-Q\rangle_F\\ &= \sum_{i,j}\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_j)-2\sum_{i,j}\alpha_i\beta_jK(\mathbf{x}_i,\mathbf{z}_j)+\sum_{i,j}\beta_i\beta_jK(\mathbf{z}_i,\mathbf{z}_j)\end{aligned}$$

作为距离计算的例子，P 必须在包含点 $\mathbf{x}_i$, $i=1,\ldots,\ell$ 的映像 $(1,\mathbf{x}_i)$ 的最小球的中心：

$$\begin{aligned}P &= (\alpha_i,\mathbf{x}_i)_{i=1}^{\ell},\text{ 这里}\\ \alpha &= \operatorname*{argmin}_{\alpha}\left(\sum_{i,j=1}^{\ell}\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_j)+\max_{1\leqslant k\leqslant\ell}\left(K(\mathbf{x}_k,\mathbf{x}_k)-2\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\mathbf{x}_k)\right)\right)\end{aligned}$$

因为从 P 到 $(1,\mathbf{x}_k)$ 的平方距离是：

$$\begin{aligned}\langle P-(1,\mathbf{x}_k)\cdot P-(1,\mathbf{x}_k)\rangle_F &= \sum_{i,j=1}^{\ell}\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_j)\\ &\quad -2\sum_{i=1}^{\ell}\alpha_iK(\mathbf{x}_i,\mathbf{x}_k)+K(\mathbf{x}_k,\mathbf{x}_k)\end{aligned}$$

右侧第一项与 k 无关。

按照这个思路，也可以使用特征向量的对偶表示在特征空间进行主成分分析。

评注 3.16　特征空间有一个有趣之处，当然不是本书主要论述的一个必要方面，但是它有助于明确输入空间映像流形的结构。上面讨论过的距离是整个特征空间的所有距离，它内在是高维的。输入空间的映像$\phi(X)$可能是一个扭曲的子流形，它的维数是输入空间的维数。在这个映像内分析距离也是可能的。沿着表面测量距离需要使用合适张量定义的黎曼度量。微分几何研究在这样的张量空间上导出的度量。度量张量是对称正定的。选择一个核导出相应的张量 g，这个张量可以通过计算两个点 $\mathbf{x}$ 和 $\mathbf{z}$ 的平方距离展开的泰勒序列的前三项决定，这两个点在表面上可以成为 $\mathbf{x}=\mathbf{z}+d\mathbf{x}$ 这样的函数。既然前两项为 0，可以得到双线性形式或平方项：

$$\begin{aligned} ds^2 &= \langle (1,\mathbf{x}) - (1,\mathbf{z}) \cdot (1,\mathbf{x}) - (1,\mathbf{z}) \rangle_F \\ &= K(\mathbf{x},\mathbf{x}) - 2K(\mathbf{x},\mathbf{z}) + K(\mathbf{z},\mathbf{z}) \\ &= \frac{1}{2}\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\left(\frac{\partial^2 K(\mathbf{x},\mathbf{x})}{\partial x_i \partial x_j}\right)_{\mathbf{x}=\mathbf{z}} dx_i dx_j - \sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\left(\frac{\partial^2 K(\mathbf{x},\mathbf{z})}{\partial x_i \partial x_j}\right)_{\mathbf{x}=\mathbf{z}} dx_i dx_j \\ &= \sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\left(\frac{1}{2}\frac{\partial^2 K(\mathbf{x},\mathbf{x})}{\partial x_i \partial x_j} - \frac{\partial^2 K(\mathbf{x},\mathbf{z})}{\partial x_i \partial x_j}\right)_{\mathbf{x}=\mathbf{z}} dx_i dx_j \end{aligned}$$

从而给出了张量 g 的成分：

$$g_{ij}(\mathbf{z}) = \left(\frac{1}{2}\frac{\partial^2 K(\mathbf{x},\mathbf{x})}{\partial x_i \partial x_j} - \frac{\partial^2 K(\mathbf{x},\mathbf{z})}{\partial x_i \partial x_j}\right)_{\mathbf{x}=\mathbf{z}}$$

因此，从核函数也能计算出黎曼度量张量，它有助于明确特征空间的结构和相应的输入空间映像的子流形。

3.5　核与高斯过程

对固定的 $\mathbf{x} \in X$，考虑函数$f(\mathbf{x})$的输出。当 f 根据定义在实值函数 $\mathscr{F}$ 上的某个分布 $\mathscr{D}$ 来选择时，可以将输出值看做随机变量，因此：

$$\{f(\mathbf{x}) : \mathbf{x} \in X\}$$

是作为潜在相关的随机变量的集合。这样的集合是一个随机过程。在函数类 $\mathscr{F}$ 上的分布可视为先验知识，在可能的情况下，不同的函数都可以为学习系统提供解。这样的先验知识就是贝叶斯学习方面的一个特性。本书将在下一章深入讨论这个方法，并在第 6 章讨论如何使用高斯过程预测。在此，希望能够说明在贝叶斯学习中特定形式的先验知识和书中介绍的 SVM 中核函数的关系。若假定分布为高斯分布，贝叶

斯分析需要的计算将会大大简化。对一个有限的变量集合$S = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$，高斯分布（零均值）由对称的正定协方差矩阵$\Sigma = \Sigma(\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$及其如下相应的分布确定：

$$P_{f\sim\mathscr{D}}[(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_\ell)) = (y_1, \ldots, y_\ell)] \propto \exp\left(-\frac{1}{2}\mathbf{y}'\Sigma^{-1}\mathbf{y}\right)$$

高斯过程是一个随机过程，其任意有限变量集合的边缘分布是零均值高斯分布。Σ的(i,j)项度量了$f(\mathbf{x}_i)$和$f(\mathbf{x}_j)$的相关性，这也是期望$E_{f\sim\mathscr{D}}[f(\mathbf{x}_i)f(\mathbf{x}_j)]$，它仅与$\mathbf{x}_i$和$\mathbf{x}_j$有关。因而存在对称协方差函数$K(\mathbf{x},\mathbf{z})$，使得$\Sigma(\mathbf{x}_1, \ldots, \mathbf{x}_\ell)_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$。协方差矩阵在所有输入点的有限集上是正定的要求和评注 3.7 给出的定义 Mercer 核的性质是吻合的，因此可以看出在空间 X 索引的变量集上定义高斯过程等价于在$X \times X$上定义 Mercer 核。用协方差函数定义高斯过程可以避免函数类$\mathscr{F}$及其先验知识的显式定义。因此，同在 SVM 中通过核隐式定义特征空间一样，函数类及其先验知识可以由高斯过程协方差函数隐式定义。定义一个函数类及其先验使核成为相应的协方差函数也是可能的，就像可以显式地为核计算特征空间一样。实际上，函数空间的一个选择是在 Mercer 特征的 F 空间中的线性函数类：

$$\mathbf{x} = (x_1, \ldots, x_n) \longmapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_j(\mathbf{x}), \ldots)$$

这些特征定义在下面加权内积定义的l_2空间 F 中：

$$\left\langle \boldsymbol{\psi} \cdot \tilde{\boldsymbol{\psi}} \right\rangle = \sum_{j=0}^{\infty} \lambda_j \psi_j \tilde{\psi}_j$$

权重向量$\boldsymbol{\psi}$上的先验分布$\mathscr{D}$是在每个坐标 i 上独立的方差为$\sqrt{\lambda_i}$的零均值高斯分布。利用这个先验知识，可以计算两个输入点 $\mathbf{x}$ 和 $\mathbf{z}$ 输出之间的相关性$C(\mathbf{x},\mathbf{z})$：

$$\begin{aligned}
C(\mathbf{x},\mathbf{z}) &= E_{\boldsymbol{\psi}\sim\mathscr{D}}\left[\langle \boldsymbol{\psi}\cdot\boldsymbol{\phi}(\mathbf{x})\rangle \langle \boldsymbol{\psi}\cdot\boldsymbol{\phi}(\mathbf{z})\rangle\right] \\
&= \int_{\mathscr{F}} \langle \boldsymbol{\psi}\cdot\boldsymbol{\phi}(\mathbf{x})\rangle \langle \boldsymbol{\psi}\cdot\boldsymbol{\phi}(\mathbf{z})\rangle \, d\mathscr{D}(\boldsymbol{\psi}) \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \phi_i(\mathbf{x})\phi_j(\mathbf{z}) \int_{\mathscr{F}} \psi_i\psi_j d\mathscr{D}(\boldsymbol{\psi}) \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \phi_i(\mathbf{x})\phi_j(\mathbf{z})\delta_{ij}\lambda_i \\
&= \sum_{i=0}^{\infty} \phi_i(\mathbf{x})\phi_i(\mathbf{z})\lambda_i = K(\mathbf{x},\mathbf{z})
\end{aligned}$$

3.6　习题

1. 在特征空间中，求出质心和点的平方距离的均值。
2. 令$K(\mathbf{x},\mathbf{z})=\exp(-\|\mathbf{x}-\mathbf{z}\|^2/\sigma^2)$是评注 3.14 中的高斯核，它可以应用到任何欧氏空间或者l_2空间。现在为输入空间X考虑在$X\times X$上的任意核$K_1(\mathbf{x},\mathbf{z})$。展示如何计算由$K_1$隐式定义的特征的高斯核，并且在$X\times X$上使用?
3. 考虑例 3.15，使用相同的特征空间为特征映射构造一个核，使得$\phi_u(s)$能够数出u作为s子串发生的次数。

3.7　补充读物和高级主题

Mercer 定理将核解释为特征空间的内积，这是 1964 年 Aizermann, Bravermann 和 Rozoener 在势函数方法的研究工作中引入机器学习领域的[1]，但是它的应用潜力直到 Boser，Guyon 和 Vapnik 在介绍 SVM[19]时才首次得到充分理解。

核的理论比较古老，Mercer 定理可以追溯到 1909 年[95]，再生核希尔伯特空间的研究是 Aronszajn 在 20 世纪 40 年代[7]开始的。这个理论在逼近和正则化理论中得到应用，可参见 Wahba 的书[171]和她在 1999 年的综述[172]。Poggio 在 1975 年首先应用了多项式核[115]。将再生核扩展用于机器学习领域和神经网络领域，是 Poggio 和 Girosi 的工作，比如他们 1990 年关于径向基神经网络的论文[116]。

正定函数的理论是在协方差和相关函数中提出的，和高斯过程的工作高度相关[180]。事实上，较早结果是在[172]的文献中得出的。Saitoh[123]显示了在评注 3.7 里提到的所有有限集的核矩阵的正性和半正定的联系。

构造核的技术可以在很多论文中找到，比如 Micchelli[97], MacKay[81], Evgeniou 等人[39], Schölkopf 等人[136], Haussler[58]和 Watkins[174]。关于 RKHS 的讨论出自于 Haussler 的论文[58]，而例 3.15 的基础出自 Watkins 的论文[176]。一维平移不变核的例 3.9 取自 Girosi[51]。特征空间的不同几何描述是 Burges[132]提供的，同时还有一些核满足 Mercer 定理的必要条件。

Schölkopf[129]，Watkins[175]和 Haussler[58]的论文极大地扩展了核的使用，显示它们实际上可以定义在一般的集合上，而不一定是欧氏空间。这拓宽了核在现实世界新的应用范围，输入空间可以是生物序列、文本、图像等。这些核推广了 Vapnik[159]描述的回归 ANOVA 核。

Joachims[67]和 Dumais 等[36]使用稀疏向量编码文本特征。Jaakkola 和 Haussler 提出使用隐马尔可夫模型计算生物序列[65]中的核，这里的特征向量是分布的 Fisher

分数。这方面的更多内容将在第 8.4.1 节详细讨论。Watkins 提出使用概率的上下文无关文法来建立序列[174]之间的核。同样，Haussler 提出对生物序列采用特殊的核[58]。

一个有趣的研究方向是直接从数据中求得核。Jaakkola[65]的论文和 Cristianini 等人[31]的论文研究了这个主题。在一篇有趣的论文里，Amari 和 Wu[3]描述了一种方法，它可以直接作用到核上来影响输入空间的几何分布，从而提高数据的分类效果。

在非线性趋势下，将核作为一种一般的技术用于线性学习器可以类推到其他学习系统，比如最近邻算法（使用第3.4节的技术）或者是Schölkopf，Smola和Mueller[134]展示的 PCA。使用特定核来处理噪声和不可分数据的技术由 Shawe-Taylor 和 Cristianini 阐述，并将在第 6 章介绍（也可以参考[140]）。

这些参考文献还会在网站 www.support-vector.net 上给出，这个网站将不断及时补充新的研究成果并提供在线软件和论文的链接。

第4章　泛化性理论

核的引入极大地提高了学习器的表达能力，并保持了内在的线性从而使得学习容易得到控制。然而适应性的增加容易导致过拟合的产生，因为随着自由度的增加分界超平面变得更加不适定。

第1章给出了学习方法内在统计推断可靠性的参考文献。成功控制核函数特征空间的适应性需要一套完善的泛化性理论，它能够精确描述控制学习器中哪个因子才能保证好的泛化能力。已经有几个学习理论可以应用于这个问题。其中，用 Vapnik 和 Chervonenkis（VC）理论描述 SVM 是最合适的，因为从历史上讲，它促进了 SVM 的出现。但是，除此之外还可以从贝叶斯角度解释 SVM。

本章将回顾 VC 理论的主要结论，其中提出了线性分类器泛化性的可靠界，由此指示如何控制核空间线性函数的复杂度。同时还将简要评述贝叶斯统计方法和压缩方法的结论，它们也可以用来描述这样的系统，并建议控制哪些参数来提高泛化能力。

4.1　可能近似正确学习模型

现在要介绍的模型在不同领域有不同的名字。在统计学中它就是所谓的一致收敛比率或频率推断；但在计算机科学中，它一般称为可能近似正确（pac, probably approximately correct）学习模型；而它在机器学习领域流行多年以前，Vapnik 和 Chervonenkis 已经将其应用于统计推断。取这个名字的原因在下面描述模型的组成时将逐渐显现。

模型的一个关键假设是训练数据和测试数据是根据某个未知但固定的分布 $\mathcal{D}$ 独立同分布(i.i.d.)产生的。假定输入/输出对$(\mathbf{x}, y) \in X \times \{-1, 1\}$是同分布，这种方法包含了输出 y 是由一个固定的目标函数 $y = t(\mathbf{x})$决定的情况。模型的修正已经考虑了当分布随时间变化或者训练集的产生不是完全无关的情况，比如样例序列作为一个时间序列产生的时候。模型还忽略了学习器影响样例选取的可能性，这在学习的查询模型中是值得研究的一个部分。本书将忽略所有这些精细之处，只考虑 i.i.d. 的情况。

既然测试样例也根据分布 $\mathcal{D}$ 产生，分类情况下自然想到的一种误差的度量就是随机产生的样例被误分的概率。进一步可以考虑正负样例的代价不相同的情况，初

始分析时将忽略这个问题。因此定义分类函数 h 在分布 $\mathcal{D}$ 上的误差 $\mathrm{err}_{\mathcal{D}}(h)$ 为：

$$\underset{\mathcal{D}}{\mathrm{err}}(h) = \mathcal{D}\{(\mathbf{x}, y) : h(\mathbf{x}) \neq y\}$$

这样的度量也称为风险函数，因为它度量了期望误差率。分析的目的是为了用几个量断定误差界。也许最关键的量是所用的训练样例个数。pac 结论通常表示为要获得特定等级误差在样本个数上的界。这也是学习问题的样本复杂度。本书倾向于用样本数目来界定误差，这个误差可以直接用做不同假设类选择的标准，也就是用于所谓模型选择问题。

考虑一个固定的推断规则，它从学习器配置的分类规则 H 中选择一个假设 h_s，这是建立在数据集

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

上，S 包括了根据分布 $\mathcal{D}$ 通过 i.i.d. 选择的 ℓ 个训练样例。在这个设置中，可以将泛化误差 $\mathrm{err}_{\mathcal{D}}(h_S)$ 看做在随机选择的训练集基础上的随机变量。泛化性分析的经典统计方法的目的是给出期望的泛化误差界，这里的期望是建立在随机选择的特定数目训练集上。许多情况下这样的估计是不可靠的，因为特定的误差可能会远离它的期望。由交叉验证技术可以给出一个不可靠的例子。学习的 pac 模型需要一个可靠的泛化误差界，因而它界定了泛化误差随机变量 $\mathrm{err}_{\mathcal{D}}(h_S)$ 分布的尾部。尾部的大小由学习器指定的参数 δ 决定。因此一个 pac 界的形式是 $\varepsilon = \varepsilon(\ell, H, \delta)$，并在随机产生的数据集 S 上至少以概率 $1-\delta$ 断定所选假设 h_S 的泛化误差界是：

$$\underset{\mathcal{D}}{\mathrm{err}}(h_S) \leqslant \varepsilon(\ell, H, \delta) \tag{4.1}$$

或者换句话说是可能近似正确学习（pac）。这等价于断定训练集在一个假设下大误差的概率满足：

$$\mathcal{D}^\ell \left\{ S : \underset{\mathcal{D}}{\mathrm{err}}(h_S) > \varepsilon(\ell, H, \delta) \right\} < \delta \tag{4.2}$$

pac 方法有统计测试的特色，它断定数据误分的概率是小的。这意味着此结论在 δ 等级上是重要的，或者测试不可靠的概率最大是 δ。在这个意义上它提供了一个统计验证的假设，这类似发展实验科学所用的方法。

pac 方法的一个关键之处在于不同于许多统计测试，其误差界与分布 $\mathcal{D}$ 无关。这意味着无论分布如何产生数据，误差界一定成立，这个性质称为分布无关。很容易理解，有一些分布的学习较其他分布困难，因此在所有分布上都成立的理论一定在许多情况下令人失望。下面将介绍大间隔方法突破了最坏情况的瓶颈，能够利用好分布的优势。我们首先介绍分布无关情况下的分析。

4.2　VC 理论

对于有限集的假设，不难获得不等式（4.1）形式的界。假定使用的推断规则是选择与训练集 S 中训练样例一致的任意假设 h。所有 ℓ 个相互无关的样例与假设 h 一致并有 $\mathrm{err}_{\mathscr{D}}(h) > \varepsilon$ 的概率界是：

$$\mathscr{D}^{\ell}\{S : \text{与 } h \text{ 一致并且 } \underset{\mathscr{D}}{\mathrm{err}}(h) > \varepsilon\} \leqslant (1-\varepsilon)^{\ell} \leqslant \exp(-\varepsilon\ell)$$

这里第二个不等式是一个简单的数学界。现在，即使假定所有 $|H|$ 个假设有大的误差，其中的一个与 S 一致的概率至多为：

$$|H|\exp(-\varepsilon\ell)$$

这是几个事件中的一个发生的概率的联合界。它界定了一致假设 h_S 误差大于 ε 的概率，就像在不等式（4.2）中给出的一样：

$$\mathscr{D}^{\ell}\{S : \text{与 } h_S \text{ 一致并且 } \underset{\mathscr{D}}{\mathrm{err}}(h_S) > \varepsilon\} < |H|\exp(-\varepsilon\ell)$$

为了确保右侧小于 δ，令：

$$\varepsilon = \varepsilon(\ell, H, \delta) = \frac{1}{\ell}\ln\frac{|H|}{\delta}$$

这个简单的界已经显示了函数类 H 通过基数度量的复杂度如何直接影响到误差界的。H 选得过大明显会引起过拟合。该结论同样显示将真实误差与在 H 的所有假设上成立的经验误差关联的性质。因此，称这里演示的是一致收敛。学习理论建立在界定误差的经验估计和真实估计的差别上，其中误差在假设集及其所设定的条件上是一致的。可以发现，假设集有限时这是不困难的。Vapnik 和 Chervonenkis 理论的主要贡献是将这样的分析扩展到无限假设集，比如前面讨论的实数权重向量的线性学习器的情况。

假定一个推断规则，由它可以得出任意一致的假设，并用 $\mathrm{err}_S(h)$ 表示假设 h 在样本集 S 上错误的次数。给出无限函数集界的关键是给出不等式（4.2）的概率界，它是在训练样本上零误差但在第二个随机样本 $\hat{S}$ 上高误差的概率的两倍：

$$\begin{aligned}\mathscr{D}^{\ell}\left\{S : \exists h \in H : \underset{S}{\mathrm{err}}(h) = 0, \underset{\mathscr{D}}{\mathrm{err}}(h) > \varepsilon\right\} \\ \leqslant 2\mathscr{D}^{2\ell}\left\{S\hat{S} : \exists h \in H : \underset{S}{\mathrm{err}}(h) = 0, \underset{\hat{S}}{\mathrm{err}}(h) > \varepsilon/2\right\}\end{aligned} \tag{4.3}$$

当 $\ell > 2/\varepsilon$ 时这个关系是 Chernoff 界的一个应用。右侧的量是通过固定 2ℓ 个样本并计数不同的次序来给出界，这些次序的点是通过控制所有误差在第二个样本选择得到。

既然每种次序是等同可能的，所以满足该性质的那部分次序的比例是其概率的上界。仅考虑交换第一个样本和第二个样本中对应点所得的次序，可以给出这样次序的比例的界为$2^{-\ell/2}$，与2ℓ个样本点的特定集合无关。在有限2ℓ个样本点上考虑误差的优势是假设空间实际上有限，因为在2ℓ个样本点上不会出现超过$2^{2\ell}$个分类函数。为了获得不等式（4.3）右侧的整体概率的联合界，所需要的是当限制到2ℓ个点时假设空间规模的界，这是一个称为生长函数的量：

$$B_H(\ell) = \max_{(\mathbf{x}_1,\ldots,\mathbf{x}_\ell)\in X^\ell} |\{(h(\mathbf{x}_1), h(\mathbf{x}_2), \ldots, h(\mathbf{x}_\ell)) : h \in H\}|$$

这里$|A|$表示集A的基数。这个量的第一个结论是它不会超过2^ℓ，因为其最大数目的集合是长度为ℓ的二值序列的所有子集。对点集$\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$有集合：

$$\{(h(\mathbf{x}_1), h(\mathbf{x}_2), \ldots, h(\mathbf{x}_\ell)) : h \in H\} = \{-1, 1\}^\ell$$

可以说点集被H打散。如果任意大小的集合都可被打散，则对于所有ℓ，生长函数等于2^ℓ。VC理论的最后一部分是关于有限d是被打散集合的最大尺寸的情况分析。在这种情况下，对$\ell \geqslant d$，生长函数的界为：

$$B_H(\ell) \leqslant \left(\frac{e\ell}{d}\right)^d$$

它用指数d给出了多项式增长的关系。d就是所谓类H的VC维，用VCdim (H)表示。该量度量了函数类的丰富性或适应性，有时也称为容量。控制学习系统的容量是提高泛化精度的一种方式。综合上述生长函数的界和对样本前半部分可以误导学习器的那部分次序的观察，可以得到不等式（4.2）左侧的界：

$$\mathscr{D}^\ell \left\{ S : \exists h \in H : \underset{S}{\mathrm{err}}(h) = 0, \underset{\mathscr{D}}{\mathrm{err}}(h) > \varepsilon \right\} \leqslant 2\left(\frac{2e\ell}{d}\right)^d 2^{-\varepsilon\ell/2}$$

从而得到了任意一致假设h的pac界：

$$\underset{\mathscr{D}}{\mathrm{err}}(h) \leqslant \varepsilon(\ell, H, \delta) = \frac{2}{\ell}\left(d\log\frac{2e\ell}{d} + \log\frac{2}{\delta}\right)$$

这里$d = \mathrm{VCdim}(H)$。因此得出下面部分的学习基本定理。

定理4.1 （Vapnik和Chervonenkis）令H是VC维为d的假设空间。对在$X \times \{-1, 1\}$上的任意概率分布$\mathscr{D}$，与ℓ个随机样本集S一致的任意假设空间$h \in H$在S上的误差以概率$1 - \delta$不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(h) \leqslant \varepsilon(\ell, H, \delta) = \frac{2}{\ell}\left(d\log\frac{2e\ell}{d} + \log\frac{2}{\delta}\right)$$

条件是$d \leqslant \ell$，$\ell > 2/\varepsilon$。

评注 4.2　这一定理显示了对于无限假设集，过拟合的问题是可以避免的，所使用的复杂度的度量正是 VC 维。要保证好的泛化能力，在一致假设情况下训练样本的大小跟这个量呈线性关系。

VC 理论为一致假设提供了分布无关下的一个泛化性界，由此还可看出更紧凑的界是 log 因子的，下面的定理更清楚地说明了这个问题。

定理 4.3　*令 H 是有限 VC 维 $d \geqslant 1$ 的假设空间。对任意学习算法，存在分布使得在 ℓ 个随机样例上至少以概率 δ，算法返回的假设 h 的误差至少为：*

$$\max\left(\frac{d-1}{32\ell}, \frac{1}{\ell}\ln\frac{1}{\delta}\right)$$

评注 4.4　定理说明了对具有高 VC 维的假设类，存在输入概率分布要求学习器增大训练集来获得好的泛化能力。可以看出有限 VC 维刻画了 pac 意义上的学习能力——将误差表达为有限 VCdim (H)的函数，而在分布无关下学习无限 VC 维是不可能的。注意此下界并不对所有分布都成立。如果分布较好，对于一个高 VC 维的函数类的学习是可能的。事实上，对于 SVM 的性能这个事实是必要的，因为 SVM 是利用好的分布的优势来设计的。这将在下一节深入讨论。

为了将这个理论应用到线性学习器，必须用维数 n 计算 $\mathbb{R}^n$ 中的线性函数类 $\mathscr{L}$ 的 VCdim($\mathscr{L}$)，它决定了能被不同的线性函数形成的所有 2^d 个可能的分类方式分开的样例的最大数目 d，也就是能被 $\mathscr{L}$ 打散的最大数目。下面的命题刻画了这种情况在什么时候发生。

命题 4.5　*令 $\mathscr{L}$ 是 $\mathbb{R}^n$ 上的线性学习器类。*

- *在一般位置（不是在 $n-1$ 维的仿射子空间）上 $n+1$ 个训练样例的任意集 S，在 $\mathscr{L}$ 中存在函数，不论 S 中训练点的标记如何均能一致分类 S。*
- *对 $\ell > n+1$ 个输入的任意集，最少存在一个分类不能被 $\mathscr{L}$ 中的任意函数实现。*

定理 4.3 和命题 4.5 意味着极高维特征空间中的学习是不可能的。一个极端的例子是高斯核应用到一个无限维的特征空间。因此可以推论出，根据分布无关的 pac 分析支持向量机的学习是不能成功的。而支持向量机能够学习的事实说明产生样例的分布并不是定理 4.3 的下界所要求的最坏情况。下一节将描述一个更精确的 pac 分析，它显示分类器的间隔提供了在辨识目标概念时分布有用性的一个度量，并产生了如下形式的泛化误差界：

$$\underset{\mathscr{D}}{\mathrm{err}}(h) \leqslant \varepsilon(\ell, \mathscr{L}, \delta, \gamma)$$

它不包括特征空间的维数。因此，SVM 学习策略能够使用分布和目标概念间的特殊

关系，这在现实世界的应用中经常发生。这种类型的界包括了度量训练过程的结果的量，称为数据相关。

这里描述的理论仅用于当假设与训练数据一致的情况。如果数据有噪声或者假设类不能捕捉目标函数的丰富性，则这是不可能的，或者需要进一步提高类的一致性。可以修正这个理论从而允许在训练集上有一定数目的错误，修正是通过计数左侧不再有更多错误的次序进行的。所得泛化误差界由下面的定理给出。

定理 4.6 *令 H 是 VC 维为 d 的假设空间。对在 $X \times \{-1,1\}$ 上的任意概率分布 $\mathcal{D}$，在 ℓ 个随机样例集 S 上误差为 k 的任意假设 $h \in H$ 在 S 上的误差以概率 $1-\delta$ 不大于：*

$$\underset{\mathcal{D}}{\mathrm{err}}(h) \leqslant \varepsilon(\ell, H, \delta) = \frac{2k}{\ell} + \frac{4}{\ell}\left(d \log \frac{2e\ell}{d} + \log \frac{4}{\delta}\right)$$

条件是 $d \leqslant \ell$。

这个定理建议假设空间 H 的训练算法应该最小化训练误差数目，因为界中其他任何量都已经被 H 所固定。这个归纳原理也就是经验风险最小化，它寻求最小化风险函数的经验度量值。这个定理也可以应用到一个嵌套序列的假设空间类：

$$H_1 \subset H_2 \subset \ldots \subset H_i \subset \ldots \subset H_M$$

它使用 δ/M，因此使得任意一个界不成立的概率小于 δ。如果在每个类 H_i 中找到最小训练误差的假设 h_i，则在固定训练集 S 上的错误数 k_i 将满足：

$$k_1 \geqslant k_2 \geqslant \ldots \geqslant k_i \geqslant \ldots \geqslant k_M$$

同时 VC 维 $d_i = \mathrm{VCdim}(H_i)$ 形成了一个递增序列。定理 4.6 的界可以用来选择使其最小的假设 h_i，它使得误差数目（第一项）的减少超过了容量（第二项）的增加。这里介绍的策略称做结构风险最小化。

4.3 泛化性的间隔界

回顾定义 2.2 中给出的分类器间隔的定义。现在将这个定义推广到任意实值函数类。

定义 4.7 *考虑在输入空间 X 上使用一实值函数类 $\mathcal{F}$ 来分类，阈值为 0。定义样例 $(\mathbf{x}_i, y_i) \in X \times \{-1,1\}$ 对应于函数 $f \in \mathcal{F}$ 的间隔是：*

$$\gamma_i = y_i f(\mathbf{x}_i)$$

注意 $\gamma_i > 0$ 意味着 $(\mathbf{x}_i, y_i)$ 正确分类。对应于训练集 S 的 f 的间隔分布是 S 中样例的间

隔分布。有时将间隔分布的最小值称为对应于训练集 S 的 f 的间隔 $m_S(f)$。如果 f 正确分类 S，这个值是正的。最后，对应于类 $\mathscr{F}$ 训练集 S 的间隔是在所有 $f \in \mathscr{F}$ 上的最大间隔。

下面三小节将分别考虑在训练集 $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$ 上，实值函数 f 的间隔分布：

$$M_S(f) = \{\gamma_i = y_i f(\mathbf{x}_i) : i = 1, \ldots, \ell\}$$

的不同度量表示的界。如果考虑一个线性函数类，可以假定间隔是几何间隔（见定义 2.2），或者说权重向量是单位范数。下面将讨论间隔 $m_S(f)$ 或最小 $M_S(f)$。

4.3.1　最大间隔界

证明定理 4.1 时将无限假设集上的概率转化到 2ℓ 样本的有限假设集上。大间隔 γ 可以缩小函数空间的有效规模，是因为泛化性能可以由双样本点上输出在 $\gamma/2$ 内的函数逼近。许多情况下，在 ℓ 个点的固定集上在 $\gamma/2$ 内逼近整个类的行为的函数集规模大大小于阈值类的生长函数的大小。有代表性样本函数的规模估计需要一些额外的机理和符号。

定义 4.8　令 $\mathscr{F}$ 是在域 X 上的一类实值函数。对应于一系列输入点：

$$S = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell)$$

$\mathscr{F}$ 的 γ 覆盖是有限函数集 B，这样对所有 $f \in \mathscr{F}$，存在 $g \in B$，满足 $\max_{1 \leqslant i \leqslant \ell} (|f(\mathbf{x}_i) - g(\mathbf{x}_i)|) < \gamma$。这样覆盖的最小尺寸可以用 $\mathscr{N}(\mathscr{F}, S, \gamma)$ 表示。$\mathscr{F}$ 覆盖个数的值是：

$$\mathscr{N}(\mathscr{F}, \ell, \gamma) = \max_{S \in X^\ell} \mathscr{N}(\mathscr{F}, S, \gamma)$$

现在，显示在训练集 S 上的一个具有间隔 $m_S(f) = \gamma$ 的假设 f，如何用潜在的实值函数类的覆盖个数来重新形成定理 4.1。假定有一个固定的阈值，$\mathrm{err}_S(f)$ 计数 f 在样本集 S 上阈值输出的错误次数。类似地，$\mathrm{err}_{\mathscr{D}}(f)$ 是在根据分布 $\mathscr{D}$ 随机产生的点上定义的。因此有：

$$\begin{aligned} & \mathscr{D}^\ell \left\{ S : \exists f \in \mathscr{F} : \underset{S}{\mathrm{err}}(f) = 0, m_S(f) \geqslant \gamma, \underset{\mathscr{D}}{\mathrm{err}}(f) > \varepsilon \right\} \\ \leqslant \; & 2\mathscr{D}^{2\ell} \left\{ S\hat{S} : \exists f \in \mathscr{F} : \underset{S}{\mathrm{err}}(f) = 0, m_S(f) \geqslant \gamma, \underset{\hat{S}}{\mathrm{err}}(f) > \frac{\varepsilon \ell}{2} \right\} \end{aligned} \tag{4.4}$$

考虑对应于序列 $S\hat{S}$ 上 $\mathscr{F}$ 的一个 $\gamma/2$ 覆盖 B，令 $g \in B$，在 f 的 $\gamma/2$ 内。然后 g 有

$\mathrm{err}_S(g)=0, m_S(g)>\gamma/2$，而如果$f$在某个点$\mathbf{x}\in\hat{S}$出现错误，那么$g$在$\mathbf{x}$上一定有小于$\gamma/2$的间隔。如果$(\gamma/2)$-$\mathrm{err}_{\hat{S}}(g)$表示$\hat{S}$中$g$的间隔小于$\gamma/2$的点的个数，类似地，可以通过次序参数和联合界给出不等式（4.4）右侧的界：

$$\begin{aligned}
& 2\mathscr{D}^{2\ell}\left\{S\hat{S}:\exists f\in\mathscr{F}:\operatorname*{err}_{S}(f)=0, m_S(f)\geqslant\gamma, \operatorname*{err}_{\hat{S}}(f)>\varepsilon\ell/2\right\} \\
\leqslant\ & 2\mathscr{D}^{2\ell}\left\{S\hat{S}:\exists g\in B:\operatorname*{err}_{S}(g)=0, m_S(g)>\gamma/2, (\gamma/2)\text{-}\operatorname*{err}_{\hat{S}}(g)>\varepsilon\ell/2\right\} \\
\leqslant\ & 2|B|2^{-\varepsilon\ell/2}\leqslant 2\mathscr{N}(\mathscr{F},2\ell,\gamma/2)2^{-\varepsilon\ell/2}
\end{aligned}$$

因而初步得到下面的结论。

定理 4.9 *考虑阈值化一个实值函数空间$\mathscr{F}$并固定$\gamma\in\mathbb{R}^+$。对在$X\times\{-1,1\}$上的任意概率密度分布$\mathscr{D}$，具有间隔$m_S(f)\geqslant\gamma$的假设$f\in\mathscr{F}$在ℓ个随机样例S上的误差以概率$1-\delta$不大于：*

$$\operatorname*{err}_{\mathscr{D}}(f)\leqslant\varepsilon(\ell,\mathscr{F},\delta,\gamma)=\frac{2}{\ell}\left(\log\mathscr{N}(\mathscr{F},2\ell,\gamma/2)+\log\frac{2}{\delta}\right)$$

条件是$\ell>2/\varepsilon$。

评注 4.10 定理显示了如何用$m_S(f)$来界定泛化误差，这个量可以从训练的结果观察到。对大的γ值期望有小的$\mathscr{N}(\mathscr{F},2\ell,\gamma/2)$。这个量可以视做一个有效 VC 维，因此可以期望观察到在小的样本上大的间隔能产生好的泛化能力。注意$\mathscr{F}$的 VC 维没有进入这个界。后面有例子显示它的 VC 维实际上是无限的，但这个有效 VC 维仍然是有限的，因此可以继续学习。这与定理 4.3 的下界不矛盾，这也是所观察到的大的间隔预示分布是好的。尽管界在所有分布上成立，但在好的分布上它才有意义。

评注 4.11 定理仅能应用到学习开始前所指定的固定γ值的情况下。为了能在训练后将定理应用于观察到的γ值，一定要将定理应用于一定范围的值中，确保有一个与训练的真实输出接近。δ以 log 因子的形式进入到界中的事实使得有限集上进行这样一个一致应用而不在界内的质量上产生可观的损失成为可能。选择γ值并获得一个一致结果的细节相当具有技术性，但主要信息很少增加。因此不再深入到这个细节中（详见第 4.8 节给出的参考文献），而是将注意力转到如何界定$\log\mathscr{N}(\mathscr{F},2\ell,\gamma/2)$，如果想使用这个结论，这是一个关键的量。

基于$\log\mathscr{N}(\mathscr{F},\ell,\gamma)$的界表示了 VC 理论所需的基于生长函数的界的推广。在那种情况下，关键的度量是 VC 维d，而生长函数是以自由度d的多项式生长的。要用来界定覆盖个数的相应的量是 VC 维的实值推广，又称为宽打散维。

定义 4.12　令 $\mathscr{F}$ 是定义在域 X 上的实值函数类。如果要说点集 $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_\ell\}\in X^\ell$ 是被 $\mathscr{F}$ 以 γ 打散，就必须存在实数 $r_i, i=1,\ldots,\ell$，对每一个二值分类 $\mathbf{b}\in\{-1,1\}^\ell$，存在 $f_\mathbf{b}\in\mathscr{F}$，满足：

$$f_\mathbf{b}(\mathbf{x}_i)\begin{cases}\geqslant r_i+\gamma, \text{如果 } \mathbf{b}_i=1\\ <r_i-\gamma, \text{如果 } \mathbf{b}_i=-1\end{cases}$$

尺度为 γ 的宽打散维 $\mathrm{fat}_\mathscr{F}(\gamma)$ 是 X 的最大 γ 打散子集的大小。

这个维又称为尺度敏感 VC 维。实数 r_i 可以看做每个点的单独阈值，而 γ 打散意味着可以用所选阈值以 γ 间隔实现分类。很明显，γ 值越大，可以打散的点集的规模越小，因为函数上的限制越来越严格。如果阈值 r_i 对所有点相同，那么这个维数也称为水平宽打散。选择单独阈值的自由度可以看做是引入了额外的适应性，但在线性函数的情况下，它不增加打散的点集的规模。在考察了下面用宽打散维表示的覆盖个数的界后，将返回到线性函数类。

引理 4.13　令 $\mathscr{F}$ 是一类函数 $X\to[a,b]$，并且 $\mathscr{D}$ 是在 X 上的一个分布。选择 $0<\gamma<1$ 并令 $d=\mathrm{fat}_\mathscr{F}(\gamma/4)$。则对 $\ell\geqslant d$ 有：

$$\log\mathscr{N}(\mathscr{F},\ell,\gamma)\leqslant 1+d\log\frac{2e\ell(b-a)}{d\gamma}\log\frac{4\ell(b-a)^2}{\gamma^2}$$

在 $\mathscr{N}(\mathscr{F},\ell,\gamma)$ 上的界比多项式的要稍微大一些，但是如果忽略 log 因子，$\mathscr{F}$ 在宽打散维上对 $\log\mathscr{N}(\mathscr{F},\ell,\gamma)$ 的依赖性刚好相当于 H 在 VC 维上对 $\log B_H(\ell)$ 的依赖性。因此当观察到 γ 间隔时，可以考虑把 $\gamma/8$ 的宽打散维视为有效 VC 维。事实上，对固定 γ 值的定理 4.9 使用引理 4.13 可以得到下面的推论。

推论 4.14　考虑在区间 $[-R,R]$ 内阈值化一个实值函数空间 $\mathscr{F}$，并固定 $\gamma\in\mathbb{R}^+$。对在 $X\times\{-1,1\}$ 上的任意概率分布 $\mathscr{D}$，在 ℓ 个随机样例 S 上具有间隔 $m_S(f)\geqslant\gamma$ 的假设 $f\in\mathscr{F}$ 在 S 上的误差以概率 $1-\delta$ 不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(f)\leqslant\varepsilon(\ell,\mathscr{F},\delta,\gamma)=\frac{2}{\ell}\left(d\log\frac{16e\ell R}{d\gamma}\log\frac{128\ell R^2}{\gamma^2}+\log\frac{4}{\delta}\right)$$

条件是 $\ell>2/\varepsilon, d<\ell$，这里 $d=\mathrm{fat}_\mathscr{F}(\gamma/8)$。

评注 4.15　注意如果忽略 log 因子，这个界中宽打散维的地位与定理 4.1 中 VC 维的地位类似，但是这个量的实际值与所观察到的间隔有关，因此表示为有效 VC 维。

再次考虑 γ 值大的变化范围，需要一些额外的技术，但这些细节不会改变整个结论，这里也不再加以讨论。更多细节参见第 4.8 节给出的参考文献。可以将不同值上的结果分层，如同将假设赋给与间隔有关的不同复杂度的类。因此，函数类是数据

相关的，不像经典的结构风险最小化必须在看到数据之前指定。因此，这种类型的结论也称为数据相关的结构风险最小化。

现在将注意力转到线性函数类的宽打散维的界，这是给出 SVM 界的最后一步。

定理 4.16 假定 X 是在内积空间$\mathbb{H}$上半径为 R 的球，$X=\{\mathbf{x}\in\mathbb{H}:\|\mathbf{x}\|_{\mathbf{H}}\leqslant R\}$，考虑函数类：

$$\mathscr{L}=\{\mathbf{x}\mapsto\langle\mathbf{w}\cdot\mathbf{x}\rangle:\|\mathbf{w}\|_{\mathbf{H}}\leqslant 1,\mathbf{x}\in X\}$$

则：

$$\operatorname*{fat}_{\mathscr{L}}(\gamma)\leqslant\left(\frac{R}{\gamma}\right)^2$$

定理的证明从两个中间结论得出。一个表述为如果 $S=\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$被 $\mathscr{L}$ 以 γ 间隔打散，则每个子集 $S_0\subseteq S$ 满足：

$$\left\|\sum S_0-\sum(S-S_0)\right\|_{\mathbf{H}}\geqslant\ell\gamma \tag{4.5}$$

这里集的加和是指包含在点集中向量的加和。这个结果是考察范数中的向量和权重向量的内积得出的，其中权重向量以间隔 γ 实现了 S_0 的分类。第二个中间结果是计算左侧在子集 S_0 的随机选择下范数平方的期望得到。如果 s 是标示 S_0 中成员的 $\{-1,1\}$ 向量，则可以均匀随机选择 s，并一定要估计：

$$\begin{aligned}E\left\|\sum S_0-\sum(S-S_0)\right\|_{\mathbb{H}}^2 &= E\left\|\sum_{i=1}^{\ell}s_i\mathbf{x}_i\right\|_{\mathbf{H}}^2\\ &= E\sum_{i=1}^{\ell}s_i^2\|\mathbf{x}_i\|_{\mathbf{H}}^2+2E\sum_{i\neq j}s_is_j\langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle_{\mathbf{H}}\\ &= E\sum_{i=1}^{\ell}\|\mathbf{x}_i\|_{\mathbf{H}}^2\leqslant R^2\ell\end{aligned}$$

既然一定至少存在一个 S_0，它的值小于或等于期望值，至少存在一个分类满足：

$$\left\|\sum S_0-\sum(S-S_0)\right\|_{\mathbf{H}}\leqslant R\sqrt{\ell}$$

由这个不等式与不等式（4.5）得出 $R\sqrt{\ell}\geqslant\ell\gamma$，并得到结论。

评注 4.17 注意线性函数学习器的宽打散维的界类似于定理 2.3 给出的感知机算法的误差界。

下面给出 SVM 的误差界。

定理 4.18　考虑在内积空间 X 上阈值化具有单位权重向量的实值线性函数 $\mathscr{L}$，并固定 $\gamma \in \mathbb{R}^+$。在 $X \times \{-1,1\}$ 上的任意概率分布 $\mathscr{D}$，在以原点为球心，半径为 R 的球内，在 ℓ 个随机样例集 S 上具有间隔 $m_S(f) \geqslant \gamma$ 的假设 $f \in \mathscr{L}$ 在 S 上的误差以概率 $1-\delta$ 不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(f) \leqslant \varepsilon(\ell, \mathscr{L}, \delta, \gamma) = \frac{2}{\ell}\left(\frac{64R^2}{\gamma^2}\log\frac{e\ell\gamma}{4R}\log\frac{128\ell R^2}{\gamma^2} + \log\frac{4}{\delta}\right)$$

条件是 $\ell > 2/\varepsilon, 64R^2/\gamma^2 < \ell$。

这个结论的一个重要的定量方面是输入空间的维数没有出现，事实上这个结论也可以用于无限维空间。这种类型的结论有时称为维数无关，因而预示界可以克服维数灾难。在第 4.2 节最后结论的基础上，可以得出维数灾难的避免只有当产生样例的分布足够好时才是可能的，这使得辨识对应的目标函数更容易一些。在这样的情况下，界以较高的概率保证在随机选取的测试样本上有小的误差。只有在这个意义上，可以将 γ 看做分布好坏的一种度量，从而可期望算法有多高的泛化能力。以测试点距超平面的距离给出误分概率更精确的估计也是可能的，细节参见第 4.8 节给出的参考文献。

对于不可分数据或数据中有噪声导致间隔很小的情况，定理 4.18 是平凡的，没有给出任何信息。下面的两小节讨论了处理这种情况的两种方法，它们使用了间隔分布的不同度量。

4.3.2　间隔百分界

间隔分布用来界定泛化性的下一个度量是通用百分点。这个度量的重要优势是它涵盖了假设与训练数据不完全一致的情况。如果将下面的分布值排序：

$$M_S(f) = \{\gamma_i = y_i f(\mathbf{x}_i)\}$$

得到 $\gamma_1 \leqslant \gamma_2 \leqslant \cdots \leqslant \gamma_\ell$，并且固定数值 $k < \ell$，$M_S(f)$ 的这个 k/ℓ 百分值 $M_{S,k}(f)$ 就是 γ_k。下面的定理用 k/ℓ 和 $M_{S,k}(f)$ 给出了一个泛化误差的界。

定理 4.19　考虑在内积空间 X 上阈值化具有单位权重向量的实值线性函数 $\mathscr{L}$，并固定 $\gamma \in \mathbb{R}^+$。有一个常数 c，使得在 $X \times \{-1,1\}$ 上的任意概率分布 $\mathscr{D}$，在以原点为球心，半径为 R 的球内，在 ℓ 个随机样例集 S 假设 $f \in \mathscr{L}$ 的误差以概率 $1-\delta$ 不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(f) \leqslant \frac{k}{\ell} + \sqrt{\frac{c}{\ell}\left(\frac{R^2}{M_{S,k}(f)^2}\log^2\ell + \log\frac{1}{\delta}\right)}$$

对所有 $k < \ell$ 成立。

定理 4.19 的证明与定理 4.9 的证明有相似的模式，只是定理 4.9 中双样本次序的计数在定理 4.19 中更加复杂，变为需要允许右侧有一些误差。

此定理建议可以通过最小化间隔误差的数目获得最好的泛化性能，如果训练点的间隔小于γ，定义训练点为γ间隔误差。这个界可以处理少数离群点的情况，而离群点会使得训练集或者不可分，或者有很小的间隔。这个定理忽略了困难点的间隔，而只使用剩余点的间隔。定理使用更大间隔的代价有两层。首先，额外项k/ℓ考虑了要忽略部分训练集的事实，其次是比定理 4.18 增加了平方根。下一节的界所使用的间隔分布不使用平方根，而是用间隔误差大小的度量来表示。

4.3.3 软间隔界

本小节开始给出间隔松弛变量的准确定义。该定义将定义 2.6 推广到一般函数类，当$\mathscr{F}$是线性函数类的时候可以简化为定义 2.6。这个定义的主旨是考虑当目标间隔是γ的时候有多少独立点达不到目标。对于间隔大于γ的点，这个数值为 0，而对于误分点，松弛变量要大于γ。

定义 4.20 考虑在输入空间 X 上使用实值函数类$\mathscr{F}$取阈值为 0 进行分类。对应于函数$f \in \mathscr{F}$ 和目标间隔γ定义样例$(\mathbf{x}_i, y_i) \in X \times \{-1,1\}$的间隔松弛变量（见图 2.4）为：

$$\xi((\mathbf{x}_i, y_i), f, \gamma) = \xi_i = \max(0, \gamma - y_i f(\mathbf{x}_i))$$

注意$\xi_i > \gamma$意味着$(\mathbf{x}_i, y_i)$不正确的分类。训练集：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

的间隔松弛向量$\boldsymbol{\xi}(S, f, \gamma)$对应着函数$f$和目标间隔$\gamma$，包含了间隔松弛变量：

$$\boldsymbol{\xi} = \boldsymbol{\xi}(S, f, \gamma) = (\xi_1, \ldots, \xi_\ell)$$

这里由于上下文清楚，S, f, γ的依赖性被略去。

数据噪声使单个点的间隔变小以至于成为负值，因此松弛变量可以作为数据噪声的度量。所以从松弛变量得来的方法适合处理噪声数据。

下面将用目标间隔γ和间隔松弛向量的不同范数导出假设f的泛化性界。技巧是将没有达到目标间隔的点移出，这是通过将输入空间嵌入到一个更大空间，在这个空间中可以找到函数增加移出点的间隔。移动的代价可以用间隔松弛向量的范数度量。对输入空间 X，使用辅助内积空间：

$$L(X) = \left\{ f \in \mathbb{R}^X : \text{supp}(f)\text{是可数的，并且} \sum_{\mathbf{x} \in \text{supp}(f)} f(\mathbf{x})^2 < \infty \right\}$$

$f, g \in L(X)$的内积由下式给出：

$$\langle f \cdot g \rangle = \sum_{\mathbf{x} \in \mathrm{supp}(f)} f(\mathbf{x}) g(\mathbf{x})$$

现在使用下面的映射将输入空间嵌入到空间$X \times L(X)$：

$$\tau : \mathbf{x} \mapsto (\mathbf{x}, \delta_{\mathbf{x}})$$

这里：

$$\delta_{\mathbf{x}}(\mathbf{z}) = \begin{cases} 1, \text{ 当 } \mathbf{x} = \mathbf{z} \\ 0, \text{ 其他} \end{cases}$$

因此，将输入$\mathbf{x}_i$映射到一个值$c_i \in \mathbb{R}$，$i = 1, 2, \ldots,$的通用函数$g \in L(X)$可以写为：

$$g = \sum_{i=1}^{\infty} c_i \delta_{\mathbf{x}_i}$$

既然$L(X)$是内积空间，可以将$L(X)$的元素看做$L(X)$上的线性函数。因此，对函数$(f, g) \in \mathcal{F} \times L(X)$，定义在$(\mathbf{x}, \phi) \in X \times L(X)$上$(f, g)$的函数为：

$$(f, g)(\mathbf{x}, \phi) = f(\mathbf{x}) + \langle g \cdot \phi \rangle$$

这样，(f, g)在$\tau(\mathbf{x})$上的函数为：

$$(f, g)(\tau(\mathbf{x})) = f(\mathbf{x}) + \langle g \cdot \delta_{\mathbf{x}} \rangle$$

这里的策略是巧妙选择$g \in L(X)$使得组合函数的间隔为γ，而扩展函数类的覆盖个数可以用间隔松弛向量的范数给出。首先显示如何选择g，使得数据可以用γ间隔分开。定义下面的辅助函数$g_f = g(S, f, \gamma) \in L(X)$：

$$g_f = \sum_{i=1}^{\ell} \xi_i y_i \delta_{\mathbf{x}_i}$$

现在对于$(\mathbf{x}_i, y_i) \in S$：

$$\begin{aligned} y_i (f, g_f)(\tau(\mathbf{x}_i)) &= y_i f(\mathbf{x}_i) + y_i \langle g_f \cdot \delta_{\mathbf{x}_i} \rangle \\ &= y_i f(\mathbf{x}_i) + y_i \sum_{j=1}^{\ell} \xi_j y_j \langle \delta_{\mathbf{x}_j} \cdot \delta_{\mathbf{x}_i} \rangle \\ &= y_i f(\mathbf{x}_i) + \xi_i y_i^2 \\ &= y_i f(\mathbf{x}_i) + \xi_i \geqslant \gamma \end{aligned}$$

因此，辅助函数在训练集上的确有间隔γ，但它对不在训练集中的点$\mathbf{x}$的作用与f一样：

$$\begin{aligned}(f,g_f)(\tau(\mathbf{x})) &= f(\mathbf{x})+\sum_{j=1}^{\ell}\xi_j y_j\left\langle \delta_{\mathbf{x}_j}\cdot\delta_{\mathbf{x}}\right\rangle \\ &= f(\mathbf{x})\end{aligned}$$

因此两个函数的泛化能力是一样的。所以有下面的结论。

定理 4.21 考虑阈值化一个实值函数空间$\mathscr{F}$并固定一个子空间$L\subseteq L(X)$和$\gamma\in\mathbb{R}^+$。在$X\times\{-1,1\}$上的任意概率分布$\mathscr{D}$，具有$g(S,f,\gamma)\in L$的任意假设$f\in\mathscr{F}$在ℓ个随机样例S上的误差以概率$1-\delta$不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(f)\leqslant\varepsilon(\ell,\mathscr{F},\delta,\gamma)=\frac{2}{\ell}\left(\log\mathcal{N}(\mathscr{F},2\ell,\gamma/4)+\log\mathcal{N}(L,2\ell,\gamma/4)+\log\frac{2}{\delta}\right)$$

条件是$\ell>2/\varepsilon$，并且在误分训练点上没有离散概率。

证明 从上面看出，$(f,g(S,f,\gamma))$在训练点上有间隔γ，而在训练集外的点上等同f。因此可以应用定理 4.9 给出不在训练集的点的界。剩下的是给出$\log\mathcal{N}(\mathscr{F}\times L,2\ell,\gamma/2)$上的一个逼近界。令 A 是$\mathscr{F}$的一个覆盖，B 是 L 的一个覆盖，它们对应于2ℓ个点$\mathbf{x}_1,\ldots,\mathbf{x}_{2\ell}$的尺度是$\gamma/4$。则$A\times B$是对于相同点的$\mathscr{F}\times L$上的一个$\gamma/2$覆盖，因为对于一般的$(f,g)\in\mathscr{F}\times L$，可以找到$f'\in A$，满足：

$$|f(\mathbf{x}_i)-f'(\mathbf{x}_i)|\leqslant\gamma/4\ \ i=1,\ldots,2\ell$$

并且有$g'\in B$，满足：

$$|g(\delta_{\mathbf{x}_i})-g'(\delta_{\mathbf{x}_i})|\leqslant\gamma/4\ \ i=1,\ldots,2\ell$$

这时，有：

$$\begin{aligned}|(f,g)(\tau(\mathbf{x}_i))-(f',g')(\tau(\mathbf{x}_i))| &\leqslant |f(\mathbf{x}_i)-f'(\mathbf{x}_i)|+|g(\delta_{\mathbf{x}_i})-g'(\delta_{\mathbf{x}_i})| \\ &\leqslant \gamma/2\ \ i=1,\ldots,2\ell\end{aligned}$$

因此得出：

$$\mathcal{N}(\mathscr{F}\times L,2\ell,\gamma/2)\leqslant\mathcal{N}(\mathscr{F},2\ell,\gamma/4)\mathcal{N}(L,2\ell,\gamma/4)$$

从而得到结论。

为了应用这个结论，必须选择$L(X)$子空间的合适序列：

$$L_1\subset L_2\subset\ldots\subset L_k\subset\ldots\subset L(X)$$

在给定γ,S,f条件下为每个子空间应用定理，然后选择包含$g(S,f,\gamma)$最小L_k。将要考虑的是用函数的二阶范数和一阶范数定义的序列。在二阶范数下，有一个内积空间，应用定理 4.16 和引理 4.13 可以给出覆盖个数的界，并获得下面的结论。

定理 4.22　考虑在内积空间 X 上阈值化具有单位权重向量的实值线性函数 $\mathscr{L}$，并固定 $\gamma \in \mathbb{R}^+$。存在常数 c，使得在 $X \times \{-1,1\}$ 上的任意概率分布 $\mathscr{D}$，在以原点为球心，半径为 R 的球内，具有任意假设 $f \in \mathscr{L}$ 在 ℓ 个随机样例 S 上以概率 $1-\delta$ 不大于：

$$\operatorname*{err}_{\mathscr{D}}(f) \leqslant \frac{c}{\ell}\left(\frac{R^2+\|\xi\|_2^2}{\gamma^2}\log^2\ell+\log\frac{1}{\delta}\right)$$

这里 $\xi=\xi(f,S,\gamma)$ 是对应于 f 和 γ 的间隔松弛向量。

评注 4.23　一个类似的界是感知机算法在第一个迭代中的误差界，这在定理 2.7 给出。

如果序列 L_k 由一阶范数项定义，那么获得的界将在这个界上附加一个额外的 log 因子。

定理 4.24　考虑在内积空间 X 上阈值化具有单位权重向量的实值线性函数 $\mathscr{L}$，并固定 $\gamma \in \mathbb{R}^+$。存在常数 c，使得在 $X \times \{-1,1\}$ 上的任意概率分布 $\mathscr{D}$，在以原点为球心，半径为 R 的球内，具有任意假设 $f \in \mathscr{L}$ 在 ℓ 个随机样例 S 上以概率 $1-\delta$ 不大于：

$$\operatorname*{err}_{\mathscr{D}}(f) \leqslant \frac{c}{\ell}\left(\frac{R^2+\|\xi\|_1^2\log(1/\gamma)}{\gamma^2}\log^2\ell+\log\frac{1}{\delta}\right)$$

这里 $\xi=\xi(f,S,\gamma)$ 是对应于 f 和 γ 的间隔松弛向量。

从定理 4.22 和定理 4.24 得出的结论中可看出，泛化误差界考虑了没有满足间隔 γ 的数据点的个数。这个界以松弛向量的范数表示，说明要优化性能，需要最小化这个量。这个界与线性可分数据点无关，因此它能够处理被噪声腐蚀的数据集或函数类没有完全捕捉到决策规则全部复杂性的情况。优化间隔松弛向量的范数不意味着要最小化误分次数。因此定理建议的归纳原则并不对应着经验风险最小化。这个事实是重要的，下面将要看到最小化误分次数所需的计算多于优化间隔松弛向量。

优化间隔松弛向量的范数在边界上有扩散效果。因此，相对于最大间隔它又称为软间隔，最大间隔只依赖训练点的一个子集，因此称为硬间隔。以后称二阶范数间隔松弛向量的界为二阶范数软间隔界，类似的包括一阶范数软间隔。

4.4　其他泛化界和幸运度函数

前一节考察了用间隔分布的度量表示的泛化性能的界。其中强调的是在高维空间使用的界一定要利用输入分布的优势以及与目标函数的关系。界一定要避免依赖输入空间的维数，而是依赖训练算法结果的度量，因为它有效地度量输入分布的有利性。前面描述了用间隔分布的三个度量表示的三个结论。本节要强调的是这种类

型的界不一定依赖间隔值。尤其是样本压缩方案也可以用来界定泛化性，这是Littlestone 和 Warmuth 通过一个相对直接的参数获得的。一个样本压缩方案可以通过下面的固定规则定义：

$$\rho : S \mapsto \rho(S)$$

这个规则从一套标记的数据中构造了一个分类器。给定一个大的训练集，它通过找到一个最小子集（压缩集）$\hat{S} \subseteq S$ 来压缩，在压缩集中重构的分类器 $\rho(\hat{S})$ 可以正确分类整个数据集 S。固定 $d < \ell$。假定对特定的训练集，获得了一个大小为 d 的压缩集。这只能有 $\binom{\ell}{d}$ 种方式。对每种选择，所得假设误差超过 ε，并且正确分类剩余 $\ell - d$ 随机产生训练点的概率的界是：

$$(1-\varepsilon)^{\ell-d} \leqslant \exp(-\varepsilon(\ell-d))$$

因此，一个大小为 d 的压缩集误差超过 ε_d 的概率可以如下界定：

$$\binom{\ell}{d} \exp(-\varepsilon_d(\ell-d)) \tag{4.6}$$

对：

$$\varepsilon_d = \frac{1}{\ell-d}\left(d\ln\frac{e\ell}{d} + \ln\frac{\ell}{\delta}\right)$$

将小于 δ/ℓ。这得出 ε_d 不能给出大小为 d 的压缩集的泛化误差界的概率小于 δ/ℓ，所以对应于压缩集所观察到的 ε 比泛化误差大的概率至多为 δ。因此有下面的定理。

定理 4.25 *考虑一个压缩方案 ρ。对 $X \times \{-1,1\}$ 上的任意概率分布 $\mathcal{D}$，大小为 d 的压缩集定义的假设在 ℓ 个随机样例 S 上的误差以概率 $1-\delta$ 不大于：*

$$\underset{\mathcal{D}}{\mathrm{err}}(f) \leqslant \frac{1}{\ell-d}\left(d\log\frac{e\ell}{d} + \log\frac{\ell}{\delta}\right)$$

第 6 章中将介绍 SVM 的支持向量形成了一个压缩方案，它可以重构最大间隔超平面。定理 4.25 显示了支持向量的个数，一个不直接包含间隔的量，可以用来度量产生数据的分布与目标函数的一致性，因此给出了另一个数据相关的界。这个观察引入了一个通用框架，它使用了幸运度函数来评价数据分布与目标函数的关系。间隔的大小只是其中一个度量。选择一个幸运度函数对应着断定了数据分布与目标函数关系类型的先验信念。如果这个信念正确，收益是提高了泛化能力，当然假设失败也会有小的代价。

4.5　回归的泛化性

回归问题就是在训练样本上找到一个函数，它可以从输入域近似映射到实数值上。输出值不再是二值，这意味着假设输出值与训练值间的偏差不再是离散的。称两个值的差为输出值的残差，它是在这个点上拟合精度的指标。需要决定如何度量精度，小的残差难以避免，但不希望有大的残差。损失函数决定了这个度量。损失函数的不同选择将导致回归策略的不同。比如，最小二乘回归使用残差平方和作为损失函数。

尽管有几种不同的途径（见第 4.8 节），本书将使用分类情况下获得的界对回归函数的泛化性能进行分析，这启发第 6 章描述的算法。因此将考虑一个阈值测试精度 θ，超出它就认为是有误差。这样的目的是提供一个随机测试点的精度少于 θ 的概率界。如果使用相同 θ 来评价训练集性能，就可以将实值回归器作为分类器，最坏的情况应用下界。为了有效使用维数无关界，必须使用一个回归器精度间隔来跟分类间隔相对应。使用同样的符号 γ 来表示这个间隔，它度量了训练集和测试集精度差别的量。应该强调的是训练和测试中使用了不同的损失函数，这里 γ 度量了不同损失函数的差别，意味着如果训练点的精度少于 $\theta-\gamma$ 就认为是有误差。评估性能的一种可视化方法是考虑假设函数两侧大小为 $\pm(\theta-\gamma)$ 的带。任何在带外的训练点被认为是误差点。$\pm\theta$ 带外的测试点将被认为是误差点。使用相应的定理 4.18 可以有下面回归估计的解释。

定理 4.26　*考虑在内积空间 X 上用单位权重向量的线性函数 $\mathcal{L}$ 进行回归估计，固定 $\gamma \leqslant \theta \in \mathbb{R}^+$。对 $X \times \mathbb{R}$ 上的任意概率分布 $\mathcal{D}$，在以原点为球心，半径为 R 的球内，在所有 ℓ 个训练样例集 S 上训练输出值，以概率 $1-\delta$ 在 $\theta-\gamma$ 之内的任意假设 $f \in \mathcal{L}$ 在随机抽取的测试点上有余值大于 θ 的概率至多：*

$$\operatorname*{err}_{\mathcal{D}}(f) \leqslant \varepsilon(\ell, \mathcal{L}, \delta, \gamma) = \frac{2}{\ell}\left(\frac{64R^2}{\gamma^2}\log\frac{e\ell\gamma}{4R}\log\frac{128\ell R^2}{\gamma^2} + \log\frac{4}{\delta}\right)$$

条件是 $\ell > 2/\varepsilon$，$64R^2/\gamma^2 < \ell$。

这个定理给出了当训练点上的残差都小于 $\theta-\gamma$ 时单位范数线性函数在随机测试点上的输出超出 θ 的概率界。注意就像在分类情况下，特征空间的维数没有出现在公式中，确保了界可以应用到随着核的使用而带来的高维特征空间。

下面考虑间隔松弛变量在回归情况下的作用。

定义 4.27　考虑在输入空间 X 上使用实值函数类 $\mathcal{F}$ 进行回归。对应于函数 $f \in \mathcal{F}$、目标精度 θ 和损失间隔 γ，定义样例 $(\mathbf{x}_i, y_i) \in X \times \mathbb{R}$ 的间隔松弛变量（图 4.1 对应于线

性函数，而图 4.2 对应于非线性函数）为：

$$\xi((\mathbf{x}_i, y_i), f, \theta, \gamma) = \xi_i = \max\left(0, |y_i - f(\mathbf{x}_i)| - (\theta - \gamma)\right)$$

注意 $\xi_i > \gamma$ 意味着 $(\mathbf{x}_i, y_i)$ 的误差超过 θ。训练集：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

的间隔松弛向量 $\boldsymbol{\xi}(S, f, \theta, \gamma)$ 对应函数 f、目标精度 θ 和损失间隔 γ，包括了间隔松弛变量：

$$\boldsymbol{\xi} = \boldsymbol{\xi}(S, f, \theta, \gamma) = (\xi_1, \ldots, \xi_\ell)$$

这里由于上下文清楚，S, f, θ, γ 的依赖性被略去。

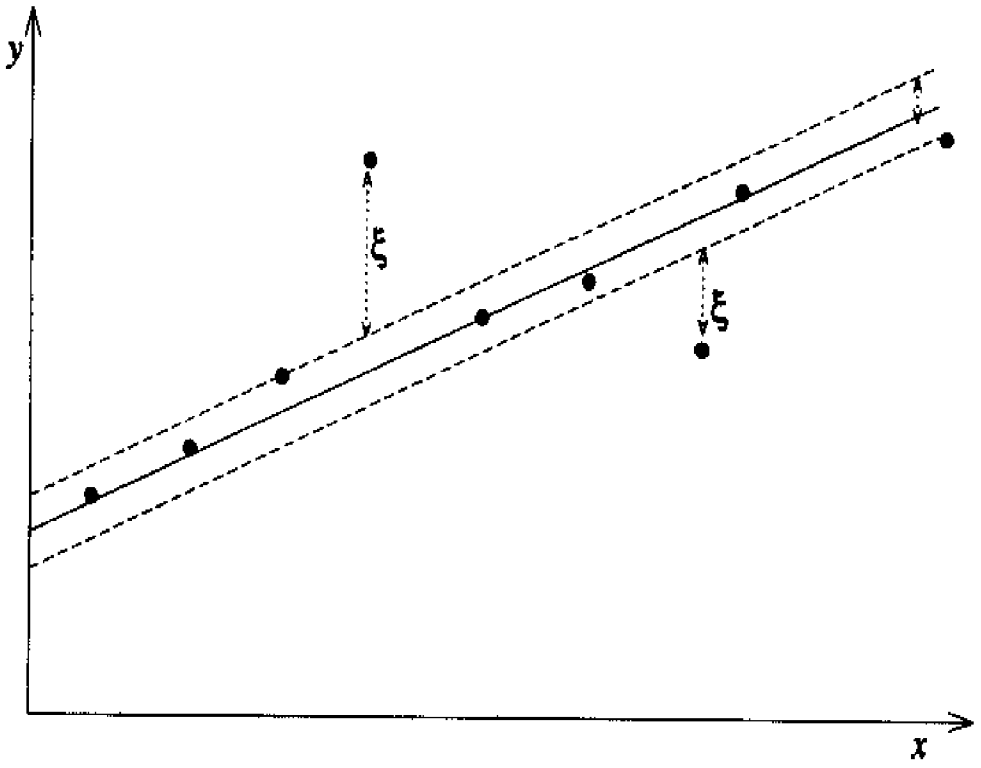

图 4.1 一维线性回归问题中的间隔松弛变量

回归和分类情况的对应是直接的，因此得出了下面线性回归器泛化性能以松弛变量的二阶范数表示的界。注意在回归情况下，固定权重向量的范数不再有意义，因为同分类情况相反的是尺度变化会引起不同的功能。权重向量范数的大小影响了尺度，因此一定要找到跟线性函数单位权重向量等价的覆盖。

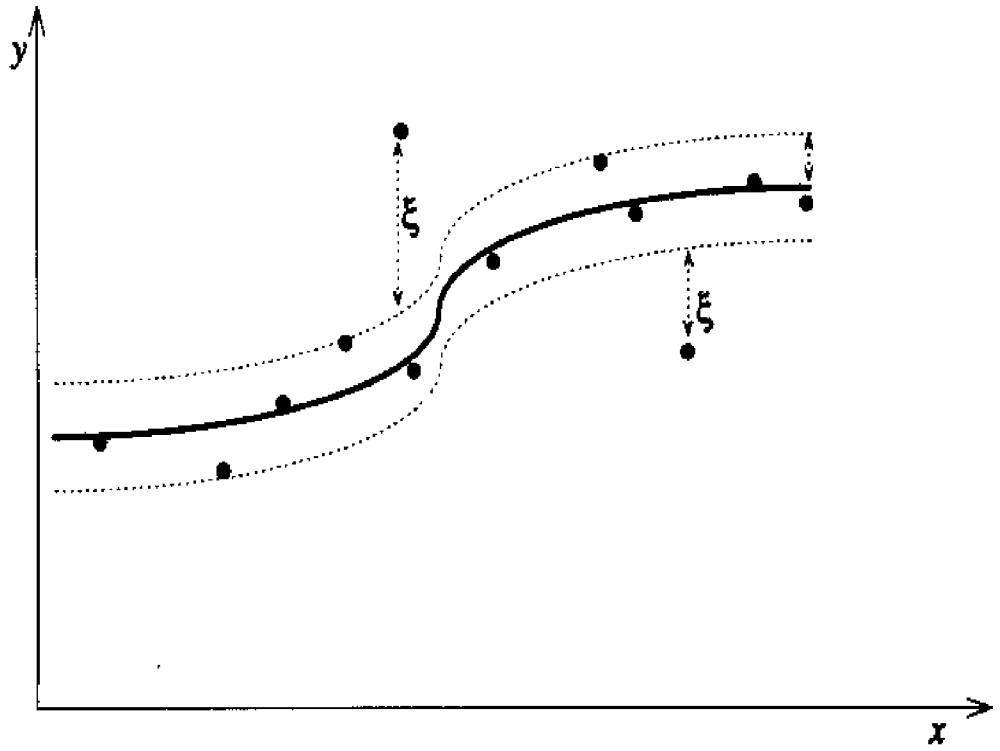

图 4.2　非线性回归函数的间隔松弛变量

定理 4.28　考虑在内积空间 X 上利用线性函数 $\mathscr{L}$ 做回归，固定 $\gamma \leqslant \theta \in \mathbb{R}^+$。存在常数 c，使得 $X \times \mathbb{R}$ 上的任意分布 $\mathscr{D}$，在以原点为球心，半径为 R 的球内，假设 $\mathrm{w} \in \mathscr{L}$ 在 ℓ 个随机样例 S 上的输出值离真实值不大于 θ 的概率以概率 $1-\delta$ 不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(f) \leqslant \frac{c}{\ell}\left(\frac{\|\mathrm{w}\|_2^2 R^2 + \|\xi\|_2^2}{\gamma^2}\log^2 \ell + \log\frac{1}{\delta}\right)$$

这里 $\xi = \xi(\mathrm{w}, S, \theta, \gamma)$ 是对应于 w，θ 和 γ 的间隔松弛向量。

这个定理比定理 4.26 有显著优势，因为它可以应用到所有线性函数，并考虑了残差在 $\theta-\gamma$ 管道之外的训练点。这些残差的二阶范数同线性函数的二阶范数一起进入了公式。如果考虑 $\gamma = \theta$ 的情况，间隔松弛向量的二阶范数就是训练序列残差的平方和，有时称为误差平方和（SSE, sum squared error）。因而有下面的推论。

推论 4.29　考虑在内积空间 X 上利用线性函数 $\mathscr{L}$ 做回归，固定 $\theta \in \mathbb{R}^+$。存在常数 c，使得在 $X \times \mathbb{R}$ 上的任意分布 $\mathscr{D}$，在以原点为球心，半径为 R 的球内，假设 $\mathrm{w} \in \mathscr{L}$ 在 ℓ 个随机样例 S 上的输出值离真实值不大于 θ 的概率以概率 $1-\delta$ 不大于：

$$\underset{\mathscr{D}}{\mathrm{err}}(f) \leqslant \frac{c}{\ell}\left(\frac{\|\mathrm{w}\|_2^2 R^2 + SSE}{\theta^2}\log^2 \ell + \log\frac{1}{\delta}\right)$$

这里 SSE 是函数 $\mathbf{w}$ 在训练集 S 上的误差平方和。

这个推论可以直接应用到使用线性函数的最小平方回归，也许是回归最标准的形式，但是这里的使用条件是训练序列根据某个未知概率分布产生的。测试点残差大于θ的概率界是评价这些函数的一种新方式。第 6 章中将介绍用第 2.2.2 节讨论的岭回归算法直接优化这个界，因此优于标准最小二乘算法。

最后变换定理 4.24 的一阶范数界得到下面的结论。

定理 4.30 *考虑在内积空间 X 上利用线性函数 $\mathcal{L}$ 做回归，固定 $\gamma \leqslant \theta \in \mathbb{R}^+$。存在常数 c，使得在 $X \times \mathbb{R}$ 上的任意分布 $\mathcal{D}$，在以原点为球心，半径为 R 的球内，假设 $\mathbf{w} \in \mathcal{L}$ 在 ℓ 个随机样例 S 上的输出值离真实值不大于 θ 的概率以概率 $1-\delta$ 不大于：*

$$\underset{\mathcal{D}}{\mathrm{err}}(f) \leqslant \frac{c}{\ell}\left(\frac{\|\mathbf{w}\|_2^2 R^2 + \|\xi\|_1^2 \log(1/\gamma)}{\gamma^2}\log^2 \ell + \log\frac{1}{\delta}\right)$$

这里 $\xi = \xi(\mathbf{w}, S, \theta, \gamma)$ 是对应于 $\mathbf{w}$，θ 和 γ 的间隔松弛向量。

这个界是以松弛向量的一阶范数和权重向量的二阶范数为项，看起来是两种不同范数的不自然的混合。然而线性函数二阶范数的使用是由函数类上的先验知识所定，而松弛变量的范数应该选择用于对腐蚀训练样例的噪声建模。如果优化一阶范数的界，可以使所得回归器较少考虑有大残差的点，这样就可以比二阶范数更好地处理离群点。

4.6 学习的贝叶斯分析

本节将简要回顾学习的贝叶斯方法。其动机不是来源于泛化界，因此看起来与本章有些脱节。尽管贝叶斯方法可以用来估计泛化性，本节仅包括可以启发学习策略的部分理论。本章前面几节介绍过的 pac 分析方式集中寻找以高概率成立的误差界。这个方法可以保守地看做以高可信度找到误差概率的界。相反，贝叶斯方法试图在训练值的基础上选择最可能的输出值。

这会得到一个不在初始假设集中的函数。但是如果限制在初始假设集，可以从中得到最优的一个。因而它能在可用数据上做出最好的选择。想做这样的计算，需要做出几个假定，包括假设集上的存在先验分布和噪声（高斯）模型。与 pac 分析对比，这些假定使得函数的选择较不可靠，因为 pac 分析的基础独立产生训练数据和测试数据的潜在概率分布是存在的。尽管有这些不同的起点，它所进行的计算和所得的函数与支持向量方法很接近。

如第3章所述，先验分布可用高斯过程描述，它选定一个协方差函数定义先验的

方式类似于 SVM 中选择核定义特征空间的方式。贝叶斯分析的目的是在观察特定数据的基础上更新分布。特定数据点与函数越不协调，函数的后验概率越低。贝叶斯公式：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

用于计算可选的概率，它假定真实函数和观察输出的误差由均值为零方差为σ^2的高斯分布产生。一旦更新完成或得到所有观察上的后验分布，学习器可以选择最可能的函数或者根据后验分布对不同函数的输出加权平均。最可能的函数通常称为最大化后验（MAP, maximum a posteriori）假设。第二个方法更有效，它为给定测试输入选择最可能输出，而没有限制所用的函数。实际上这个函数总是在原始假设集的凸闭包内，因为它是这些函数根据后验分布的加权平均。高斯过程中，两种方法是一致的，因为线性函数类等价于自己的凸闭包，并且后验分布是高斯的。

本书只考虑回归，因为误差高斯分布的假定只在这种情况下有意义。为了与前面一致，这里使用$y_i, i = 1,\ldots,\ell$表示训练集被噪声腐蚀的输出值。目标潜在的真实输出值用$t_i, i = 1,\ldots,\ell$表示。遗憾的是这个符号在一些文献中含义刚好相反。向量 y 和 t 假定为高斯的，它们的关系是：

$$P(\mathbf{y}|\mathbf{t}) \propto \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{t})'\Omega^{-1}(\mathbf{y}-\mathbf{t})\right]$$

这里$\Omega = \sigma^2\mathbf{I}$。像上面提到的，贝叶斯分析的目的是给定新的输入 x 和训练集 S 计算真实输出 t 的概率分布，其训练集 S，将分为输入向量矩阵 X 和对应的输出向量 y。因此，希望估计$P(t|\mathbf{x},S)$，尤其是找到哪里可以达到最大值。首先使用贝叶斯方程：

$$\begin{aligned} P(t,\mathbf{t}|\mathbf{x},S) &= P(t,\mathbf{t}|\mathbf{y},\mathbf{x},\mathbf{X}) = \frac{P(\mathbf{y}|\mathbf{t},\mathbf{x},\mathbf{X})P(t,\mathbf{t}|\mathbf{x},\mathbf{X})}{P(\mathbf{y}|\mathbf{x},\mathbf{X})} \\ &= \frac{P(\mathbf{y}|\mathbf{t})P(t,\mathbf{t}|\mathbf{x},\mathbf{X})}{P(\mathbf{y}|\mathbf{x},\mathbf{X})} \propto P(\mathbf{y}|\mathbf{t})P(t,\mathbf{t}|\mathbf{x},\mathbf{X}) \end{aligned}$$

这里将分母看做一个常数，因为它与所假设的选择无关。一旦训练集和测试点已知，它不再变化。表达式的第二个因子是，在对训练集的输出值没有了解的情况下，给定输入集真实输出值的先验分布。它通过选择协方差函数来由高斯过程决定。第一个因子是训练集输出值辨识得到的特定假设的权重。这个权重完全取决于假设输出和训练集输出的差异。留给贝叶斯学习器的任务是在 t 上做间隔，它意味着在参数 t 所有可能的取值上对 t 取特定值的概率加和。高斯分布的优势是所得分布也是高斯型，它的平均值和方差都可以求得解析解，因此可以给出 t 的最大值和预测精度的误差条。第 6 章将描述这个计算，并将所得的决策规则同 SVM 中得到的做比较。

4.7 习题

1. 证明命题 4.5。
2. 描述定理 4.9 如何能够应用到一切γ值的情况，试指出界被弱化时的结果。
3. 在第 4.4 节考虑选择ε_d使得满足：

$$\sum_{i=1}^{\ell} \delta_i = 1$$

的一些δ_d使公式（4.6）小于δ_d。写出定理 4.25 结论的推广。给定得到 d 个支持向量的概率是p_d，δ_d的哪种选择会在已推广的定理中给出最好的界。

4.8 补充读物和高级主题

在 VC 维基础上对泛化能力的分析是 Vapnik 和 Chervonenkis 从 20 世纪 60 年代开始的[162]。VC 理论已经应用到不同的领域，比如促进了大数一致定律的统计学的发展[117]。多数的基本理论结果已经在 Vapnik 的著作[157]中发表。机器学习 pac 模型的发展从另一方面可以追溯到 Valiant 在 1984 年的学术论文[155]，它奠定了计算学习理论的基础，描述了大量模型，包括：在线学习、查询学习、监督学习和无监督学习，并且最近应用到了强化学习。在这个理论中 VC 维结论和下界的引入源自于 Blumer 等人的一篇标志性的论文[18]，并极大影响了机器学习领域。VC 维理论用来分析学习系统的性能，比如决策树、神经网络和其他，机器学习实际应用中的许多学习启发和原理都是以 VC 理论解释的。

有许多计算学习理论导论性的著作，比如[6,71]，但它们主要集中在 pac 模型上，有时会忽略了在线、查询和非监督学习这些丰富的领域。模式识别统计分析的一个导读是 Devroye 等人给出的[33]。VC 理论最近也出现在统计学习理论中，在 Vapnik 最近的著作中[159]广泛描述，并出现在以它为导言的著作中[157,158]，当然也出现在 Vapnik 和 Chervonenkis 的早期论文中[162,164,165]。对这个理论的一个简单介绍是在[169]中。计算学习理论每年举办的国际会议为 COLT 会议，会上将探讨新的研究结果。www.neurocolt.org 网站提供了最近论文的数据库。

许多研究者提出过间隔如何影响泛化性的问题，比如 Duda 和 Hart[35]，Vapnik 和 Chervonenkis[166]以及 Mangasarian。Vapnik 和 Chervonenkis[163]得到的界使用了一个类似宽打散维的量并且是在训练集和测试集的组合上度量间隔得出的。宽打散维（有时称为尺度敏感维，或V_γ维）隐含在一些早期的参考文献中，论文[72]将其介绍到计算学习理论中，在 Alon 等人[2]的论文中特征化了 Glivenko Cantelli 类。不同

的作者得到了线性分类器的宽打散维[54,138]，本章的证明取自[9]。

得出大间隔结论的第一篇论文是[138]，[10]和另一篇参考文献包含了第 4.3.2 节中的百分界。包括分类和回归的间隔松弛向量的软间隔界出现在[141,142,139,140]中，并使用了类似第2 章讨论的技术获得了不可分情况下的误差界（进一步的参考文献见第 2.5 节）。[9,149]总结了这些结果。与间隔松弛向量相关的量是所谓的“铰链损失”，用来计算在线学习框架[48]的误差界。间隔分析也应用到 Adaboost[127]、贝叶斯分类器[32]、决策树[12]、神经网络[10]等系统中，是机器学习的标准工具。间隔分析也扩展到考虑测试样本的间隔中[137]。

Anthony 和 Bartlett 使用宽打散维在回归上获得的结果类似定理 4.26。泛化性的不同分析在回归上是可能的，比如[159]。著作[5]是回归分析方面很优秀的介绍资料。

间隔分析需要 VC 理论中的不同工具，就是因为用来刻画假设类的丰富性的量是间隔，它与数据有关。只有训练完学习器后，才能知道所得假设的复杂性。这种方式的分析提供了一种充分利用目标函数和输入分布的方式，也就是数据相关分析，或者数据相关结构风险最小化。第一个数据相关结论是关于压缩方案泛化能力的定理 4.25，这归功于 Littlestone 和 Warmuth[79,42]，而论文[138]引入了第 4.4 节提到的一般幸运度框架。其他数据相关结论包括微选择算法和 pac 贝叶斯界[93,94]。更多的结论界包括[133,37]，如同[138]一样指出了分类和回归的联系。

贝叶斯分析是统计学中的一个传统领域，应用到模式识别中已有很长时间[35]。最近几年，对贝叶斯分析的一个新的高潮从神经网络界发起，主要归功于 MacKay 的著作[82]。这种分析的介绍可以参阅 Bishop[16]和 Neal[102]的著作。最近，研究注意力集中在高斯过程，它是统计学中的一个标准工具，在[120,180]中进行了描述。本书将在第 6 章返回到高斯过程的讨论。高斯过程泛化性的贝叶斯分析由 Sollich[150]以及 Opper 和 Vivarelli[106]完成。泛化性的其他分析是可能的，可以在统计学（例如，见[105]）或是在线算法理论[75]的基础上实现。

这些参考文献也在网站 www.support-vector.net 上给出，这个网站将不断及时补充新的研究成果并提供在线软件和论文的链接。

第 5 章　最优化理论

第 4 章中出现的所有推导策略都有类似的形式，就是选择一个假设函数来最小（或最大）化一个特定的函数。在线性学习器（LLM, Linear Learning machine）中，这就相当于寻找一个参数向量，通常该向量在某种约束下使某个代价函数最小（或最大）。最优化理论是数学上的一个分支，主要是刻画此类问题的解并开发有效的算法来找到这些解。因此，机器学习的问题就转化为一种可以在最优化理论框架中进行分析的形式。

根据特定的代价函数和约束的本质，可以得出许多类最优化问题，这些问题已经过深入研究并找到很有效的解决方法。本章将描述一些结论，这些结论是在将最优化理论应用到代价函数是凸二次函数，约束是线性函数的情况下得出的。这类最优化问题称为凸二次规划，而这类问题的解决方法对于处理 SVM 的训练问题是足够的。

最优化理论不仅提供算法技术，并且将一个给定函数的充要条件定义为一个解。比如对偶理论，它提供了一个前面章节提到的 LLM 的对偶表示形式的自然解释。此外，对于解的数学结构的深入理解也启发了第 7 章中提到的许多特定算法的启发式方法和实现技术。

5.1　问题的形成

本章中所考察的问题的一般形式是寻找函数在某些约束条件下的最大值或最小值。一般的最优化问题描述如下：

定义 5.1　（原始最优化问题）给定在域 $\Omega \subseteq \mathbb{R}^n$ 上定义的函数 f, g_i, $i = 1,\ldots,k$ 与 h_i, $i = 1,\ldots,m$：

$$\begin{array}{lll} \text{minimise} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \text{subject to} & g_i(\mathbf{w}) \leqslant 0 & i = 1,\ldots,k \\ & h_i(\mathbf{w}) = 0 & i = 1,\ldots,m \end{array}$$

这里 $f(\mathbf{w})$ 称为目标函数，剩下的关系分别称为不等式约束和等式约束。目标函数的最优值称为最优化问题的值。

为了简化描述，用 $\mathbf{g}(\mathbf{w}) \leqslant \mathbf{0}$ 来表示 $g_i(\mathbf{w}) \leqslant 0$，$i = 1,\ldots,k$。表达式 $\mathbf{h}(\mathbf{w}) = \mathbf{0}$ 与

等式约束具有相同的含义。

既然最大化问题可以转化为最小化的问题，只需改变 $f(\mathbf{w})$ 的符号，因此选择最小化并不意味着一种限制。类似地，任何约束都可重写为以上的形式。

可行区域是指目标函数的约束满足的区域，可以表示为：

$$R = \{\mathbf{w} \in \Omega: \mathbf{g}(\mathbf{w}) \leqslant \mathbf{0}, \mathbf{h}(\mathbf{w}) = \mathbf{0}\}$$

最优化问题的解是指存在一个点 $\mathbf{w}^* \in R$，使得不存在其他的点 $\mathbf{w} \in R$ 满足 $f(\mathbf{w}) < f(\mathbf{w}^*)$。这个点也称为全局最小值。如果 $\exists \varepsilon > 0$ 满足 $f(\mathbf{w}) \geqslant f(\mathbf{w}^*)$，并且 $\forall \mathbf{w} \in \Omega$ 满足 $\|\mathbf{w} - \mathbf{w}^*\| < \varepsilon$，那么点 $\mathbf{w}^* \in \Omega$ 称为 $f(\mathbf{w})$ 的局部最小。

对目标函数和约束的本质的不同假设会产生不同的优化问题。

定义 5.2　对于目标函数，等式或不等式约束都是线性函数的问题称为线性规划问题。对于目标函数是二次的，而约束都是线性函数下的最优化问题称为二次规划问题。

如果解 $\mathbf{w}^*$ 满足 $g_i(\mathbf{w}^*) = 0$，不等式约束 $g_i(\mathbf{w}) \leqslant 0$ 称为积极的（或紧的），否则称为非积极的。在这种意义上，等式约束都是积极的。有时为了向等式约束传递不等式约束，引入称为松弛变量的参数，表示为 ξ，如下式所示：

$$g_i(\mathbf{w}) \leqslant 0 \iff g_i(\mathbf{w}) + \xi_i = 0 \quad \text{其中} \xi_i \geqslant 0$$

与积极约束相关联的松弛变量为 0，而对于非积极约束预示着约束中有很大的“疏松度”。

本章将考虑有限类别的最优化问题。下面首先给出凸函数和凸集的含义。

定义 5.3　对于 $\mathbf{w} \in \mathbb{R}^n$，如果 $\forall \mathbf{w}, \mathbf{u} \in \mathbb{R}^n$ 并且对于任意的 $\theta \in (0,1)$ 有：

$$f(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \leqslant \theta f(\mathbf{w}) + (1-\theta) f(\mathbf{u})$$

实值函数 $f(\mathbf{w})$ 称为凸函数，如果不等式关系严格成立，则该函数称为严格凸函数。如果一个二次可导函数的 Hessian 矩阵是半正定的，则该函数是凸的。仿射函数是指可用某个矩阵 $\mathbf{A}$ 和向量 $\mathbf{b}$ 表示为如下形式的函数：

$$f(\mathbf{w}) = \mathbf{A}\mathbf{w} + \mathbf{b}$$

注意仿射函数有零 Hessian 矩阵，所以它是凸函数。如果 $\forall \mathbf{w}, \mathbf{u} \in \Omega$ 并对任何 $\theta \in (0,1)$，点 $(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \in \Omega$，则集合 $\Omega \subseteq \mathbb{R}^n$ 称为是凸的。

如果一个函数 f 是凸的，目标函数为 f 的无约束最优化问题的局部最小值 $\mathbf{w}^*$ 也是全局最小值，因为对于任意 $\mathbf{u} \neq \mathbf{w}^*$，根据局部最小值的定义，存在充分接近 1 的 θ 使得：

$$\begin{aligned} f(\mathbf{w}^*) &\leqslant f(\theta\mathbf{w}^* + (1-\theta)\mathbf{u}) \\ &\leqslant \theta f(\mathbf{w}^*) + (1-\theta)f(\mathbf{u}) \end{aligned}$$

从而得出$f(\mathbf{w}^*) < f(\mathbf{u})$。凸函数的这个性质使得当函数和集合是凸的时候最优化问题是可解的。

定义 5.4 如果一个最优化问题的集合Ω、目标函数和所有的约束都是凸的，则称其为是凸的。

为了训练 SVM，只考虑线性约束、凸二次目标函数的问题，并且$\Omega = \mathbb{R}^n$，因此可以考虑处理凸二次规划问题。

最优化理论既涉及到一些刻画最优点的基本性质，也涉及到求最优解的算法技术的设计。本章将集中在理论分析方面，第 7 章则探讨算法的实现问题。下一节将介绍拉格朗日（Lagrange）乘子技术及其扩展，并将其限制在二次规划问题之内。

5.2 拉格朗日理论

拉格朗日理论的最初目的是刻画没有不等式约束的最优化问题的解。这个理论的主要概念是拉格朗日乘子和拉格朗日函数。这个方法是 Lagrange 总结了 1629 年 Fermat 的一个结论，在 1797 年为解决力学问题而提出的。1951 年，Kuhn 和 Tucker 在 Kuhn-Tucker 理论中进一步将这个方法扩展到不等式约束的情况。这三个逐步扩充的结论为优化 SVM 提供了有效的方法。为了便于理解，首先介绍最容易理解的情况，然后再介绍一些更复杂的问题。当没有约束的时候，目标函数的稳态足以刻画解的特征。

定理 5.5 (Fermat) $\mathbf{w}^*$成为$f(\mathbf{w})$，$f \in C^1$最小值的必要条件是$\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}} = 0$。这个条件加上$f$是凸函数，也是一个充分条件。

下面将给出这种最优化问题的一个简单例子，该例子取自第3章，其中考虑了在再生核希尔伯特空间寻找最优逼近的问题。

例 5.6 假设在下面的一个训练集上做回归：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)) \subset (X \times Y)^\ell \subset (\mathbb{R}^n \times \mathbb{R})^\ell$$

该训练集由目标函数$t(\mathbf{x})$产生。假定例 3.11 用对偶表示：

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

则为了最小化误差的 RKHS 范数，必须最小化：

$$-2\sum_{i=1}^{\ell}\alpha_i y_i+\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)$$

核 K 的半正定性保证了目标函数是凸的。根据定理 5.5，求对应于 α_i 的导数并令其等于零，得到：

$$-2y_i+2\sum_{j=1}^{\ell}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)=0\quad i=1,\ldots,\ell$$

或者

$$\mathbf{G}\boldsymbol{\alpha}=\mathbf{y}$$

这里使用 $\mathbf{G}$ 表示 Gram 矩阵，其项为 $\mathbf{G}_{ij}=K(\mathbf{x}_i,\mathbf{x}_j)$。因此，解的参数 $\boldsymbol{\alpha}^*$ 可由下式得到：

$$\boldsymbol{\alpha}=\mathbf{G}^{-1}\mathbf{y}$$

在有约束问题中，应该定义一个拉格朗日函数，该函数融合了目标函数和约束的信息，而且它的稳态可用来求解。确切地讲，拉格朗日函数定义为目标函数加上约束的线性组合，其中的组合系数称为拉格朗日乘子。

定义 5.7　给定一个最优化问题，其中目标函数是 $f(\mathbf{w})$，等式约束 $h_i(\mathbf{w})=0$，$i=1,\ldots,m$，定义拉格朗日函数：

$$L(\mathbf{w},\boldsymbol{\beta})=f(\mathbf{w})+\sum_{i=1}^{m}\beta_i h_i(\mathbf{w})$$

其中系数 β_i 称为拉格朗日乘子。

如果一个点 $\mathbf{w}^*$ 对于一个只有等式约束的最优化问题是一个局部最小点，可能 $\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}}\neq\mathbf{0}$，但是沿着能减少 f 值的方向移动会违反一个或几个约束。为了遵守等式约束 h_i，必须沿着垂直于 $\frac{\partial h_i(\mathbf{w}^*)}{\partial \mathbf{w}}$ 的方向移动，而且为了遵守所有的约束，必须沿着用下式张成的子空间 V 垂直移动：

$$\left\{\frac{\partial h_i(\mathbf{w}^*)}{\partial \mathbf{w}}:i=1,\ldots,m\right\}$$

如果 $\frac{\partial h_i(\mathbf{w}^*)}{\partial \mathbf{w}}$ 是线性无关的，没有合法的移动可改变目标函数的值，只要 $\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}}$ 位于子空间 V 中或者换句话说当存在 β_i 使得：

$$\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}} + \sum_{i=1}^{m} \beta_i h_i(\mathbf{w}^*) = 0$$

这个观察形成了等式约束条件下最优化问题的第二个最优化结论的基础。

定理 5.8 （拉格朗日）在 $f, h_i \in C^1$ 下，对于一些 $\boldsymbol{\beta}^*$ 值，点 $\mathbf{w}^*$ 在约束 $h_i(\mathbf{w}) = 0$，$i = 1, \ldots, m$ 下是 $f(\mathbf{w})$ 的最小值的必要条件是：

$$\begin{aligned} \frac{\partial L(\mathbf{w}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{w}} &= \mathbf{0} \\ \frac{\partial L(\mathbf{w}^*, \boldsymbol{\beta}^*)}{\partial \beta} &= 0 \end{aligned}$$

如果 $L(\mathbf{w}, \boldsymbol{\beta}^*)$ 是 $\mathbf{w}$ 的凸函数，上述条件也是充分的。

两个条件中的第一个给出了一个新的方程组，而第二个返回等式约束。应用这些条件（联立地解这两个方程组）可以得到解。

例 5.9 （给定表面积的最大体积盒）考虑计算一个盒子的三个边长 w, u, v，它们使得盒子的体积最大而表面积等于一个定值 c。该问题可改写为：

$$\begin{aligned} &\text{minimise} && -wuv \\ &\text{subject to} && wu + uv + vw = c/2 \end{aligned}$$

这个问题的拉格朗日形式是 $L = -wuv + \beta(wu + uv + vw - c/2)$，最优化问题的必要条件由约束和平稳条件提供：

$$\begin{aligned} \frac{\partial L}{\partial w} &= -uv + \beta(u + v) = 0 \\ \frac{\partial L}{\partial u} &= -vw + \beta(v + w) = 0 \\ \frac{\partial L}{\partial v} &= -wu + \beta(w + u) = 0 \end{aligned}$$

这些条件意味着 $\beta v(w - u) = 0$ 和 $\beta w(u - v) = 0$，它们惟一的非平凡解是 $w = u = v = \sqrt{\frac{c}{6}}$。既然这些条件对于最小值是必要的而且平凡解有零体积，因此具有最大体积的盒子是一个立方体。

例 5.10 （最大熵分布）一个有限集 $\{1, 2, \ldots, n\}$ 上的概率分布 $\mathbf{p} = (p_1, \ldots, p_n)$ 的熵，定义为 $H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log p_i$，这里很自然 $\sum_{i=1}^{n} p_i = 1$。具有最大熵的分布可用拉格朗日函数：

$$L(\mathbf{p}, \beta) = \sum_{i=1}^{n} p_i \log p_i + \beta \left(\sum_{i=1}^{n} p_i - 1 \right)$$

通过求解一个最优化问题来解决，其中拉格朗日函数定义在域 $\Omega = \{\mathbf{p} : p_i \geqslant 0, i = 1, \ldots, n\}$。平稳条件意味着对于所有的 i 有 $\log ep_i + \beta = 0$，也表明所有的 p_i 都等于 $\frac{2^{\beta}}{e}$。将此与约束联合可以得到 $\mathbf{p} = (\frac{1}{n}, \ldots, \frac{1}{n})$。因为 Ω 是凸的，约束是仿射函数并且目标函数也是凸函数，所以它是一个项为 $(ep_i \ln 2)^{-1}$ 的对角 Hessian 矩阵，这些都表明均匀分布有最大的熵。

评注 5.11　注意如果用 $h_i(\mathbf{w}) = b_i$ 替换第 i 个约束，并考虑目标函数在最优解 $f^* = f(\mathbf{w}^*)$ 的值是 b_i 的函数，则 $\left[\frac{\partial f^*}{\partial b_i}\right]_{b_i=0} = \beta_i^*$。因此拉格朗日乘子包含了给定约束的解的灵敏度信息。

评注 5.12　注意既然约束都等于零，拉格朗日函数在最优点的值等于目标函数的值：

$$L(\mathbf{w}^*, \boldsymbol{\beta}^*) = f(\mathbf{w}^*)$$

现在考虑最一般的情况即最优化问题既包含等式约束又包含不等式约束。首先，给出广义拉格朗日的定义。

定义 5.13　给定一个在域 $\Omega \subseteq \mathbb{R}^n$ 上的最优化问题：

$$\begin{array}{lll} \text{minimise} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \text{subject to} & g_i(\mathbf{w}) \leqslant 0 & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0 & i = 1, \ldots, m \end{array}$$

定义广义拉格朗日函数为：

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{w}) \\ &= f(\mathbf{w}) + \boldsymbol{\alpha}' \mathbf{g}(\mathbf{w}) + \boldsymbol{\beta}' \mathbf{h}(\mathbf{w}) \end{aligned}$$

现在可以定义拉格朗日对偶问题。

定义 5.14　定义 5.1 中原问题的拉格朗日对偶问题如下：

$$\begin{array}{ll} \text{maximise} & \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \text{subject to} & \boldsymbol{\alpha} \geqslant \mathbf{0} \end{array}$$

这里 $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\mathbf{w} \in \Omega} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$，目标函数在最优解的值称为问题的值。

下面先证明弱对偶定理，该定理给出原问题和对偶问题的基本关系并有两个有价值的推论。

定理 5.15　令 $\mathbf{w} \in \Omega$ 是定义 5.1 中原问题的一个可行解，而 $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ 是定义 5.14 中对偶问题的可行解。则 $f(\mathbf{w}) \geqslant \theta(\boldsymbol{\alpha}, \boldsymbol{\beta})$。

证明 对于$\mathbf{w} \in \Omega$，从$\theta(\alpha, \beta)$的定义出发，有：

$$\begin{aligned}\theta(\alpha, \beta) &= \inf_{\mathbf{u} \in \Omega} L(\mathbf{u}, \alpha, \beta) \\ &\leqslant L(\mathbf{w}, \alpha, \beta) \\ &= f(\mathbf{w}) + \alpha' \mathbf{g}(\mathbf{w}) + \beta' \mathbf{h}(\mathbf{w}) \leqslant f(\mathbf{w})\end{aligned} \tag{5.1}$$

因为 $\mathbf{w}$ 的可行性意味着 $\mathbf{g}(\mathbf{w}) \leqslant \mathbf{0}$ 和 $\mathbf{h}(\mathbf{w}) = \mathbf{0}$，同时$(\alpha, \beta)$的可行性意味着 $\alpha \geqslant \mathbf{0}$。

推论 5.16 对偶问题的值的上界由原问题的值给出：

$$\sup\{\theta(\alpha, \beta) : \alpha \geqslant \mathbf{0}\} \leqslant \inf\{f(\mathbf{w}) : \mathbf{g}(\mathbf{w}) \leqslant \mathbf{0}, \mathbf{h}(\mathbf{w}) = \mathbf{0}\}$$

推论 5.17 如果$f(\mathbf{w}^*) = \theta(\alpha^*, \beta^*)$，其中$\alpha^* \geqslant \mathbf{0}$，并且$\mathbf{g}(\mathbf{w}^*) \leqslant \mathbf{0}, \mathbf{h}(\mathbf{w}^*) = \mathbf{0}$，则 $\mathbf{w}^*$和(α^*, β^*)分别是原问题和对偶问题的解。在这种情况下$\alpha_i^* g_i(\mathbf{w}^*) = 0, i = 1, \ldots, k$。

证明 既然值都是相等的，则等式（5.1）中的一系列不等式必定成为等式。特别地，如果对所有 i，$\alpha_i^* g_i(\mathbf{w}^*) = 0$，则最后的不等式只能是等式。

评注 5.18 因此，如果试图一前一后求解原问题和对偶问题，可能发现可以通过比较原问题的解和对偶问题的解之差找到问题的解。如果将这个差赋零，就可以找到最优解。这个方法需要原问题的解和对偶问题的解有相同的值，有时这得不到普遍的保证。原问题的解和对偶问题的解的值之差称为对偶间隙。

另一个检测对偶间隙消失的方法是鞍点的出现。原问题的拉格朗日函数的鞍点是一个三元组：

$$(\mathbf{w}^*, \alpha^*, \beta^*) \quad \text{其中} \mathbf{w}^* \in \Omega, \alpha^* \geqslant \mathbf{0}$$

对于所有的$\mathbf{w} \in \Omega, \alpha \geqslant \mathbf{0}$，满足另外的性质：

$$L(\mathbf{w}^*, \alpha, \beta) \leqslant L(\mathbf{w}^*, \alpha^*, \beta^*) \leqslant L(\mathbf{w}, \alpha^*, \beta^*)$$

注意这里的$\mathbf{w}$并不要求满足等式或不等式约束。

定理 5.19 对于原问题，三元组$(\mathbf{w}^*, \alpha^*, \beta^*)$是拉格朗日方程的鞍点，当且仅当其组成是原问题和对偶问题的最优解并且没有对偶间隙时，原问题和对偶问题的值为：

$$f(\mathbf{w}^*) = \theta(\alpha^*, \beta^*)$$

现在提出强对偶定理，该定理保证对于将讨论的最优化问题的原问题和对偶问题有相同的值。

定理 5.20 （强对偶定理）给定域$\Omega \subseteq \mathbb{R}^n$上的一个最优化问题：

$$\begin{array}{lll} \textit{minimise} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \textit{subject to} & g_i(\mathbf{w}) \leqslant 0 & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0 & i = 1, \ldots, m \end{array}$$

其中 g_i 和 h_i 是仿射函数，也就是对于某个矩阵 $\mathbf{A}$ 和向量 $\mathbf{b}$ 有：

$$\mathbf{h}(\mathbf{w}) = \mathbf{A}\mathbf{w} - \mathbf{b}$$

其对偶间隙为零。

现在介绍 Kuhn-Tucker 定理，该定理给出了一般的最优化问题有最优解的条件。

定理 5.21　（Kuhn-Tucker 定理）给定一个定义在凸域 $\Omega \subseteq \mathbb{R}^n$ 上的最优化问题：

$$\begin{array}{lll} \textit{minimise} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \textit{subject to} & g_i(\mathbf{w}) \leqslant 0 & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0 & i = 1, \ldots, m \end{array}$$

其中 $f \in C^1$ 是凸的，并且 g_i, h_i 是仿射函数，一般地，一个点 $\mathbf{w}^*$ 是最优点的充要条件是存在 $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ 满足：

$$\begin{aligned} \frac{\partial L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{w}} &= \mathbf{0} \\ \frac{\partial L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} &= 0 \\ \alpha_i^* g_i(\mathbf{w}^*) &= 0 \quad i = 1 \ldots, k \\ g_i(\mathbf{w}^*) &\leqslant 0 \quad i = 1 \ldots, k \\ \alpha_i^* &\geqslant 0 \quad i = 1 \ldots, k \end{aligned}$$

评注 5.22　第三个关系称为 Karush-Kuhn-Tucker 互补条件。它意味着对于积极约束有 $\alpha_i^* \geqslant 0$，但是对于非积极约束有 $\alpha_i^* = 0$。进一步，可以表明对于积极约束将 0 替换为 b_i，则 $\alpha_i^* = \left[\frac{\partial f^*}{\partial b_i}\right]_{b_i=0}$，这样拉格朗日乘子表示最优值对约束的灵敏度。扰动非积极约束对于最优化问题的解没有影响。

评注 5.23　一种解释上面结果的方式是解点可以是对应子不等式约束的两个位置中的一个，要么在非积极约束的可行区域的内部，要么在积极约束定义的边界上。第一种情况下，约束的最优化条件由 Fermat 定理给出，因此 α_i 为零。在第二种情况下，可以使用非零 α_i 的拉格朗日定理。因此 Karush-Kuhn-Tucker 条件表明要么约束是积极的，意味着 $g_i(\mathbf{w}^*) = 0$，要么相应的乘子满足 $\alpha_i^* = 0$。以上可归纳为等式 $g_i(\mathbf{w}^*)\alpha_i^* = 0$。

5.3　对偶性

使用拉格朗日定理解凸最优化问题可以使用一个对偶表示替代描述，该对偶问

题通常比原问题更容易处理，因为直接处理不等式约束是困难的。对偶问题通过引入又称为对偶变量的拉格朗日乘子来解。对偶方法来源于将对偶变量作为问题的基本未知量的思想。

可以通过把拉格朗日函数对于各个原变量的导数置零，并将得到的关系式代入原拉格朗日函数，将原问题转换为对偶问题并去除了原变量的相关性。这对应于显式地计算函数：

$$\theta(\alpha,\beta)=\inf_{\mathbf{w}\in\Omega}L(\mathbf{w},\alpha,\beta)$$

得到的函数只包含对偶变量，并且在更简单的约束条件下最大化。这个策略将在下一章采用，并已成为支持向量机理论的标准技术。所得到的原变量表达式的一个诱人的特点是它和第 2 章中介绍的对偶表示形式是严格匹配的，而且在使用它的上下文中显得很自然。因此，支持向量机中使用对偶表示形式不仅可以像第 3 章一样在高维空间中工作，并且铺平了从最优化理论得到算法技术的道路。进一步的例子是对偶间隙可以作为迭代技术的收敛条件。更多深入的结论将会从 SVM 凸二次规划问题中涌现出来。Karush-Kuhn-Tucker 互补条件意味着只有积极约束有非零对偶变量，这也意味着一些最优化问题实际变量数目要比全部训练集的规模小很多。后面将使用支持向量这个术语指那些对偶变量非零的训练样例。

例 5.24 （二次规划）下面把对偶性应用到凸二次目标函数的一个重要而特殊的场合，这是对偶性的一种实际应用：

$$\begin{array}{ll}\text{minimise} & \frac{1}{2}\mathbf{w}'\mathbf{Q}\mathbf{w}-\mathbf{k}'\mathbf{w}\\ \text{subject to} & \mathbf{X}\mathbf{w}\leqslant\mathbf{c}\end{array}$$

其中 $\mathbf{Q}$ 是一个 $n\times n$ 正定矩阵，$\mathbf{k}$ 是一个 n 维向量；$\mathbf{c}$ 是一个 m 维向量，$\mathbf{w}$ 为未知，而 $\mathbf{X}$ 是 $m\times n$ 的矩阵。假设可行区域非空，则这个问题可重写为：

$$\max_{\alpha\geqslant 0}\left(\min_{\mathbf{w}}\left(\frac{1}{2}\mathbf{w}'\mathbf{Q}\mathbf{w}-\mathbf{k}'\mathbf{w}+\alpha'(\mathbf{X}\mathbf{w}-\mathbf{c})\right)\right)$$

在 $\mathbf{w}$ 上的最小值是无约束的，在 $\mathbf{w}=\mathbf{Q}^{-1}(\mathbf{k}-\mathbf{X}'\alpha)$ 上得到。将该式代入原问题，得到其对偶形式：

$$\begin{array}{ll}\text{maximise} & -\frac{1}{2}\alpha'\mathbf{P}\alpha-\alpha'\mathbf{d}-\frac{1}{2}\mathbf{k}'\mathbf{Q}\mathbf{k}\\ \text{subject to} & \alpha\geqslant\mathbf{0}\end{array}$$

这里 $\mathbf{P}=\mathbf{X}\mathbf{Q}^{-1}\mathbf{X}'$，$\mathbf{d}=\mathbf{c}-\mathbf{X}\mathbf{Q}^{-1}\mathbf{k}$。因此，二次规划的对偶形式是另一个二次规划问题，但是其约束更为简单。

5.4　习题

1. 中心点是 $\mathbf{v}$ 半径为 R 的球体集合是：

$$B_R(\mathbf{v}) = \{\mathbf{u} : \|\mathbf{u} - \mathbf{v}\| \leqslant R\}$$

将寻找包含下述向量集合的最小半径的球的问题表示为最优化问题：

$$S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$$

图 5.1 是一个简单的二维例子。将得到的问题转化为对偶形式，因此表明解可以表达为集合 S 的线性组合而且能在一个核特征空间中求解。

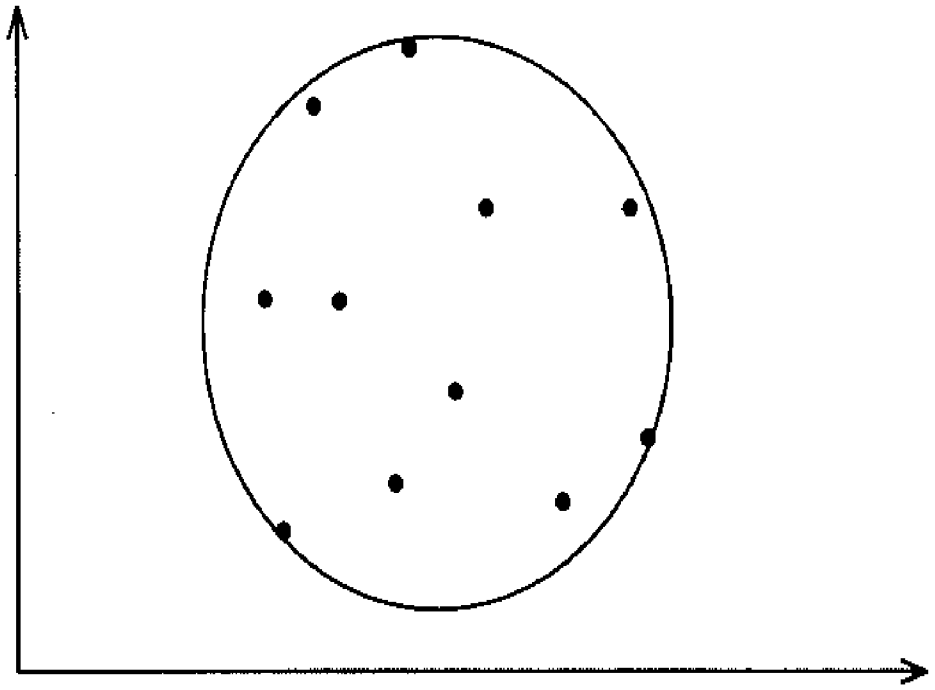

图 5.1　二维空间中包含点集的最小球（圆）的示例

2. 集合：

$$T = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$$

的凸壳是 T 中的点的所有凸组合的集合。给定一个包含正样例和负样例的线性可分训练集 $S = S^+ \cup S^-$，将在凸壳 S^+ 和 S^- 中寻找点 $\mathbf{x}^+$ 和 $\mathbf{x}^-$ 使其距离 $\|\mathbf{x}^+ - \mathbf{x}^-\|$ 最小的问题表示为一个最优化问题。注意这个距离是训练集 S 间隔的两倍。

3. 考虑一个线性学习器的权值向量参数空间。在这个空间中的每个点对应于固定基的一个假设。每个训练样例 $\mathbf{x}$ 在这个空间产生一个由以下方程定义的超

平面：

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle = 0$$

图 5.2 显示了有三个样例的二维权重向量的情况。每个超平面将空间分为能正确分类的假设和不能正确分类的假设。能够正确分类所有训练样本的假设也称为变型空间。在图 5.2 中就是居中的三角形。将寻找完全包含于变型空间的最大球的中心如图 5.2 的点 SV 的问题表达为最优化问题。注意这个点和图中变型空间的质心是不同的。将这个问题转化为对偶形式。

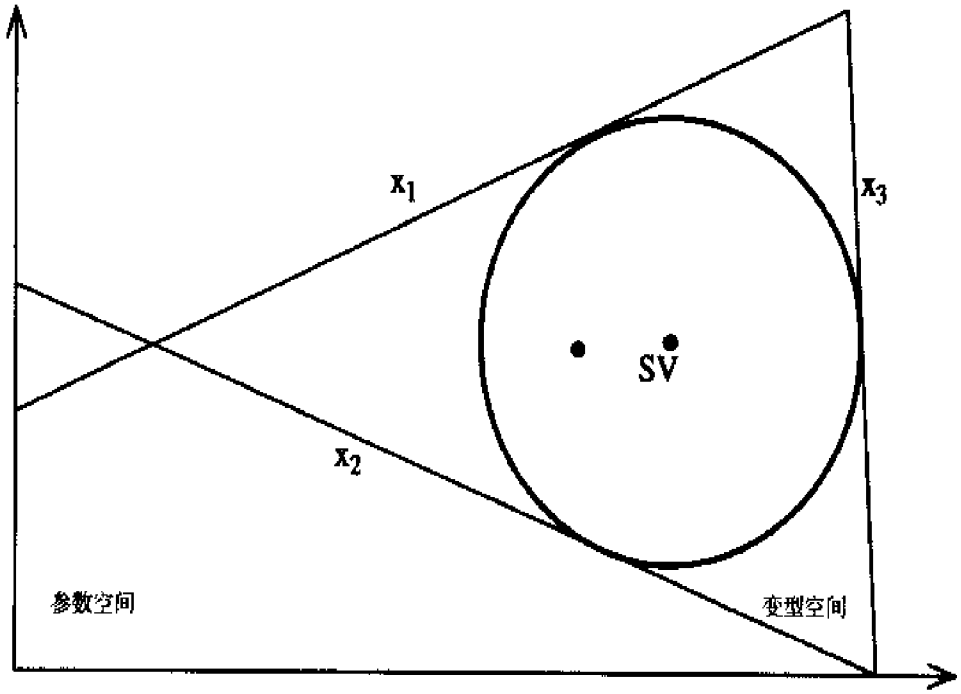

图 5.2 线性学习器的变型空间

5.5 补充读物和高级主题

最优化理论可以追溯到 Fermat 的著作，他在 17 世纪阐明无约束问题平稳下的结果。1788 年拉格朗日将其扩展到等式约束条件下。直到 1951 年 Kuhn 和 Tucker[77] 将这个理论归纳为不等式约束的情况，并提出了近代凸最优化理论。Karush 已经在 1939 年的论文[69]中描述了这个最优化条件，这就是为什么从 Kuhn-Tucker 定理中得到的条件称为 Karush-Kuhn-Tucker 条件的原因。

在其后的几年中，Wolfe，Mangasarian，Duran 和其他人做了大量工作将线性规划的对偶结果扩展到凸规划的情况（可以参考[80]的导言）。20 世纪 60 年代计算机的大量应用增加了人们对于当时称为数学规划问题的兴趣，它使用（通常线性或二次）

最优化方法来研究问题的解。

Olvi Mangasarian（比如，[84，85，87]）最先将最优化理论应用到机器学习中，他的著作是论述线性规划器。也可参见 Bennett 等人[14]将线性和二次最优化问题应用到模式识别问题中的很有价值的讨论。Mangasarian 把他的想法发挥到了极致，设计了在包含上万个点的数据集上进行数据挖掘的算法（参见第 7.8 节中更多的参考文献）。第 2 章中介绍的感知机算法也可以认为是一个简单的最优化过程，在数据给出的一套线性约束下，寻找一个可行点。但这并不是选取可行区域的任意点（一个不适定问题），而是选择一些满足极端性质的特殊点，这些性质就像第 4 章中讨论的那样，比如离可行区域边界最远。这种讨论启发了下一章支持向量机的产生。Mangasarian 的早期著作主要集中在最小化解 w 的一阶范数的问题上。最后要注意的是，将最优化的思想应用到最小二乘回归这类问题（已在第 2 章中讨论过）上，已经是这些概念在机器学习中的一种应用。

最优化理论是一个发展完善和稳定的学术领域，本章总结的标准结论可以在任何一本好的教科书中找到。一本可读性强并且全面的最优化理论的著作是[11]，经典的教科书[80，41，86]给出了很好的入门介绍。最优化理论通常也包括求解实际问题的算法和技术。本书将在第 7 章中阐述这个问题。最优化理论对于支持向量机有重要贡献，但不是在算法方面，而在于它用数学刻画了解的特性，通过 Karush-Kuhn-Tucker 条件，给出对偶变量（已在第 2 章中介绍）和间隔松驰向量（已在第 4 章介绍）的数学意义，并且一般地给出对偶问题的几何的直觉认识。

如果希望通过最优方式移动原点来最小化给点集的宽打散维数的估计，练习 1 是相关的，该练习是寻找包含数据点的最小球体的中心。这个问题最先在[128]中研究，而关于凸壳间距离的练习 2 在[14]中讨论，并给出了一个有效的方法来刻画最大间隔超平面。[73]中讨论了这个问题，它启发了一个有趣的算法（参见第 7.8 节所列的更多参考文献）。

这些参考文献也在网站 www.support-vector.net 上给出，这个网站将不断及时补充新的研究成果并提供在线软件和论文的链接。

第6章　支持向量机

前面五章的材料已经奠定了一个基础，本章将在这个基础之上介绍支持向量机，这是由 Vapnik 和他的合作者共同提出的一套学习算法。这套算法可以在第3章介绍的核特征空间中有效地训练第2章介绍的线性学习器。同时它还考虑到第4章的泛化性理论，并且使用了第5章的最优化理论。算法的一个重要特征是：根据泛化性理论强化学习偏置时，会产生假设的稀疏对偶表示，从而成为非常有效的算法。这归功于 Karush-Kuhn-Tucker 条件，这个条件保证了解，并在实际的算法实现和系统分析中占有主要地位。支持向量算法的另一个重要特征归功于核的 Mercer 条件，它使得相应的优化问题成为凸问题，因此没有局部最小。

这个特征和非零参数减少的特征使得这套算法和其他模式识别算法（比如神经网络）有明显不同。本章还将描述利用高斯过程实现贝叶斯学习策略的优化方法。

6.1　支持向量分类

支持向量分类的目的是开发计算有效的途径，从而能在高维特征空间中学习“好”的分类超平面。这里通过“好”的超平面可以理解第4章里描述的优化泛化界，而“有效计算”意味着算法能处理的样本数目在100000数量级上。泛化性理论清楚地说明了如何控制容量，因此通过控制超平面的间隔度量可以抑制过拟合，而最优化理论提供了必要的数学技术来找到优化这些度量的超平面。不同的泛化性界启发了不同的算法，比如优化最大间隔、间隔分布或支持向量的数目等。本章将考虑最通用和容易建立的方法，这些方法将问题压缩简化为一个最小化权重向量的范数问题。本章最后将提供其他相关算法的链接，因本领域的研究仍在进行，所以本书不可能包罗一切。

6.1.1　最大间隔分类器

支持向量机中最简单也是最早提出的模型是最大间隔分类器。它只能用于特征空间中线性可分的数据，因此不能在现实世界的许多情况下使用。不用说，它是最容易理解的算法，并且是更加复杂的支持向量机算法的主要模块。它展示了这类学

习器的关键特征，因此其描述对理解后面更高级的系统至关重要。

第 4 章的定理 4.18 给出了线性学习器的泛化误差界，这个界是用对应于训练集 S 的假设 f 的间隔 $m_S(f)$ 来描述的。用于分开数据的特征空间的维数没有出现在定理中。最大间隔分类器通过用分开数据的最大间隔超平面来优化这个界，并表明这个界不依赖于空间的维数，因此这个分开面可以在任何核特征空间中搜索得到。最大间隔分类器形成了第一个支持向量机的策略，从名字上看就是在一个巧妙选定的核特征空间中寻找最大间隔超平面。

要实现策略需要将其简化为凸优化问题：最小化一个线性不等式约束的二次函数。首先线性分类器的定义中有一个内在的自由度，就是即使这个超平面做尺度变换 $(\lambda\mathbf{w}, \lambda b)$，其中 $\lambda \in \mathbb{R}^+$，超平面 $(\mathbf{w}, b)$ 关联的函数也不会变化。然而相对于几何间隔而言，用函数输出的间隔会有变化。函数输出的间隔可以称为函数间隔。定理 4.18 包括了几何间隔，它是归一化权重向量后的函数间隔。因此固定函数间隔等于 1（函数间隔为 1 的超平面有时称为正则超平面），这等价于优化几何间隔，并最小化权重向量的范数。如果 $\mathbf{w}$ 是权重向量，要在正点 $\mathbf{x}^+$ 和负点 $\mathbf{x}^-$ 上实现函数间隔为 1，可以如下计算几何间隔。回顾函数间隔为 1 意味着：

$$\begin{aligned} \langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b &= +1 \\ \langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b &= -1 \end{aligned}$$

同时，为计算几何间隔必须归一化 $\mathbf{w}$。几何间隔 γ 是所得分类器的函数间隔：

$$\begin{aligned} \gamma &= \frac{1}{2}\left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|_2} \cdot \mathbf{x}^+ \right\rangle - \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|_2} \cdot \mathbf{x}^- \right\rangle \right) \\ &= \frac{1}{2\|\mathbf{w}\|_2}\left(\langle \mathbf{w} \cdot \mathbf{x}^+ \rangle - \langle \mathbf{w} \cdot \mathbf{x}^- \rangle \right) \\ &= \frac{1}{\|\mathbf{w}\|_2} \end{aligned}$$

因此几何间隔将成为 $1/\|\mathbf{w}\|_2$，并得出下面的结论。

命题 6.1　给定一个线性可分训练样本：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

求解优化问题：

$$\begin{array}{ll} \textit{minimise}_{\mathbf{w},b} & \langle \mathbf{w} \cdot \mathbf{w} \rangle \\ \textit{subject to} & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) \geqslant 1 \\ & i = 1, \ldots, \ell \end{array}$$

可以得到超平面 $(\mathbf{w}, b)$，它实现了几何间隔为 $\gamma = 1/\|\mathbf{w}\|_2$ 的最大间隔超平面。

现在考虑如何采用第 5.3 节的策略将优化问题转化为相应的对偶问题，原始拉格朗日函数为：

$$L(\mathbf{w},b,\boldsymbol{\alpha})=\frac{1}{2}\langle\mathbf{w}\cdot\mathbf{w}\rangle-\sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b\right)-1\right]$$

这里 $\alpha_i\geqslant 0$ 是拉格朗日乘子，在第 5 章已进行过论述。

通过对相应的 $\mathbf{w}$ 和 b 求偏导，可以找到相应的对偶形式：

$$\begin{aligned}\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial\mathbf{w}} &= \mathbf{w}-\sum_{i=1}^{\ell}y_i\alpha_i\mathbf{x}_i=\mathbf{0}\\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{\ell}y_i\alpha_i=0\end{aligned}$$

将得到的关系式：

$$\begin{aligned}\mathbf{w} &= \sum_{i=1}^{\ell}y_i\alpha_i\mathbf{x}_i\\ 0 &= \sum_{i=1}^{\ell}y_i\alpha_i\end{aligned}$$

代入到原始拉格朗日函数，得到：

$$\begin{aligned}L(\mathbf{w},b,\boldsymbol{\alpha}) &= \frac{1}{2}\langle\mathbf{w}\cdot\mathbf{w}\rangle-\sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b\right)-1\right]\\ &= \frac{1}{2}\sum_{i,j=1}^{\ell}y_iy_j\alpha_i\alpha_j\langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle-\sum_{i,j=1}^{\ell}y_iy_j\alpha_i\alpha_j\langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle+\sum_{i=1}^{\ell}\alpha_i\\ &= \sum_{i=1}^{\ell}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{\ell}y_iy_j\alpha_i\alpha_j\langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle\end{aligned}$$

评注 6.2 第一个替换显示假设可以描述为训练点的线性组合：应用优化理论自然地导出了第 2 章介绍的对偶表示。在应用核函数的过程中需要对偶表示。

因而上面展示了下面命题的主要部分，它沿着命题 6.1 进行。

命题 6.3 *考虑一个线性可分训练样本：*

$$S=((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_\ell,y_\ell))$$

并假定参数 $\boldsymbol{\alpha}^*$ 是下面的二次优化问题的解：

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0 \\ & \alpha_i \geqslant 0 \;\; i = 1, \ldots, \ell \end{array} \tag{6.1}$$

则权重向量 $\mathbf{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* \mathbf{x}_i$ 实现了几何间隔为：

$$\gamma = 1 / \|\mathbf{w}^*\|_2$$

的最大间隔超平面。

评注 6.4　b 的值没有出现在对偶问题中，利用原始约束一定可以找到 b^*：

$$b^* = -\frac{\max_{y_i=-1} \left(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle \right) + \min_{y_i=1} \left(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle \right)}{2}$$

第 5 章的定理 5.21 应用于这个优化问题。Karush-Kuhn-Tucker 互补条件提供了关于解的结构的有用信息。条件要求最优解 $\boldsymbol{\alpha}^*, (\mathbf{w}^*, b^*)$ 必须满足：

$$\alpha_i^* \left[y_i \left(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^* \right) - 1 \right] = 0 \qquad i = 1, \ldots, \ell$$

这意味着仅仅是函数间隔为 1 的输入点 $\mathbf{x}_i$，也就是最靠近超平面的点对应的 α_i^* 非零。所有其他点对应的参数 α_i^* 为零。因此在权重向量的表达式中，只有这些点包括在内。这就是称为支持向量的原因。图 6.1 采用大写黑体标记支持向量。

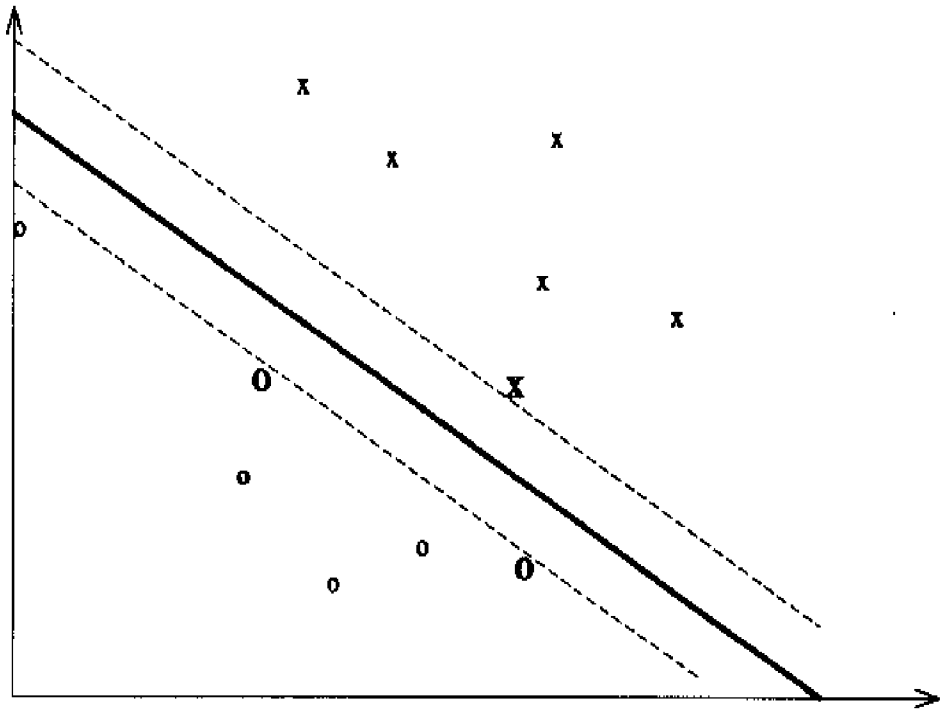

图 6.1　用大写黑体表示支持向量的最大间隔超平面

这样优化超平面就可以在对偶表示中用参数子集来表示：

$$\begin{aligned} f(\mathbf{x},\alpha^*,b^*) &= \sum_{i=1}^{\ell} y_i\alpha_i^* (\mathbf{x}_i\cdot\mathbf{x}) + b^* \\ &= \sum_{i\in sv} y_i\alpha_i^* (\mathbf{x}_i\cdot\mathbf{x}) + b^* \end{aligned}$$

与每个点关联的拉格朗日乘子成为对偶变量，并赋予了它们一个直观的解释，而且定量给出了每个训练点在所得解中的重要性。不是支持向量的点没有影响，在未退化的情况下，这些点轻微的扰动不影响解。在感知机学习算法的对偶解中可以找到类似的含义，其中对偶变量正比于训练中假设在给定点上的出错次数。

Karush-Kuhn-Tucker 互补条件的另一个重要结果在于对 $j \in sv$：

$$y_j f(\mathbf{x}_j,\alpha^*,b^*) = y_j\left(\sum_{i\in sv} y_i\alpha_i^* (\mathbf{x}_i\cdot\mathbf{x}_j) + b^*\right) = 1$$

因而

$$\begin{aligned} (\mathbf{w}^*\cdot\mathbf{w}^*) &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^*\alpha_j^* (\mathbf{x}_i\cdot\mathbf{x}_j) \\ &= \sum_{j\in sv} \alpha_j^* y_j \sum_{i\in sv} y_i\alpha_i^* (\mathbf{x}_i\cdot\mathbf{x}_j) \\ &= \sum_{j\in sv} \alpha_j^* (1 - y_j b^*) \\ &= \sum_{i\in sv} \alpha_i^* \end{aligned}$$

因此有下面的命题。

命题 6.5 考虑一个线性可分的训练样本：

$$S = ((\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_\ell, y_\ell))$$

假定参数 α^* 和 b^* 是对偶优化问题（6.1）的解，则权重向量 $\mathbf{w} = \sum_{i=1}^{\ell} y_i\alpha_i^*\mathbf{x}_i$ 实现了几何间隔为：

$$\gamma = 1/\|\mathbf{w}\|_2 = \left(\sum_{i\in sv} \alpha_i^*\right)^{-1/2}$$

的最大间隔超平面。

对偶目标函数和决策函数有一个显著的特性，就是数据仅出现在内积中。就像

在下面的命题中一样，使用核使得在特征空间中找到并使用超平面成为可能。

命题 6.6　考虑一个训练样本：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

它在核 $K(\mathbf{x},\mathbf{z})$ 隐式定义的特征空间中是线性可分的，假定参数 α^* 和 b^* 是下面的二次优化问题的解：

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0 \\ & \alpha_i \geqslant 0 \;\; i = 1, \ldots, \ell \end{array} \qquad (6.2)$$

则决策规则由 $\mathrm{sgn}(f(\mathbf{x}))$ 给出，这里 $f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$ 等价于核 $K(\mathbf{x},\mathbf{z})$ 隐式定义的特征空间中的最大间隔超平面，并且超平面有几何间隔：

$$\gamma = \left(\sum_{i \in sv} \alpha_i^* \right)^{-1/2}$$

注意核满足 Mercer 条件的要求等价于项为 $\left(K(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j=1}^{\ell}$ 的矩阵在所有训练集上是正定的要求。因此这意味着优化问题（6.2）是凸的，因为矩阵 $\left(y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j=1}^{\ell}$ 也是正定的。所以定义特征空间的核函数要求的性质确保最大间隔优化问题有惟一解并能有效得到。这跳出了神经网络训练过程中遭遇的局部最小问题。

评注 6.7　最大间隔分类器由定理 4.18 促成，它用间隔和球心在原点包含原始数据的球半径表示了泛化误差界。使用这个定理促成算法的优势是可以用学习算法的输出来计算泛化界。命题 6.6 给出了间隔的值，在特征空间中球心在原点的球半径可以这样计算：

$$R = \max_{1 \leqslant i \leqslant \ell} \left(K(\mathbf{x}_i, \mathbf{x}_i)\right)$$

遗憾的是，尽管定理建议的策略很有效率，但包含的常数使得计算所得界的实际值脱离实际。然而仍可以使用界选择不同的核函数，因为这里相对大小是重要的，现在界的准确率仍是正在研究的主题。

最优化理论一章的重要结果是定理 5.15，它显示原始目标总是比对偶目标的值要大。既然正在考虑的问题满足定理 5.20 的条件，在最优解上没有对偶间隙。因而可以使用原始值和对偶值的任意差别作为收敛的标示。这种差别称为可行间隙。令 $\hat{\boldsymbol{\alpha}}$ 是对偶变量的当前值。权重向量可以通过设置拉格朗日函数的偏导为 0 来计算，给定 $\hat{\boldsymbol{\alpha}}$ 最小化 $L(\mathbf{w}, b, \hat{\boldsymbol{\alpha}})$ 可以得到权重向量 $\hat{\mathbf{w}}$ 的当前值。因此，差别可以计算如下：

$$\begin{aligned} W(\hat{\alpha}) - \frac{1}{2}\|\hat{\mathbf{w}}\|^2 &= \inf_{\mathbf{w},b} L(\mathbf{w},b,\hat{\alpha}) - \frac{1}{2}\|\hat{\mathbf{w}}\|^2 \\ &= L(\hat{\mathbf{w}},b,\hat{\alpha}) - \frac{1}{2}\|\hat{\mathbf{w}}\|^2 \\ &= -\sum_{i=1}^{\ell}\hat{\alpha}_i\left[y_i\left(\left(\hat{\mathbf{w}}\cdot\mathbf{x}_i\right)+b\right)-1\right] \\ &= \sum_{i=1}^{\ell}\hat{\alpha}_i - \sum_{i,j=1}^{\ell}\hat{\alpha}_i y_i y_j \hat{\alpha}_j \left(\mathbf{x}_j\cdot\mathbf{x}_i\right) \end{aligned}$$

它是 Karush-Kuhn-Tucker 互补条件的加和的负数。注意这对应着原始解和对偶解的差，前提是 $\hat{\mathbf{w}}$ 满足原始约束，即假定对所有 i 有 $y_i\left(\left(\hat{\mathbf{w}}\cdot\mathbf{x}_i\right)+b\right)\geqslant 1$，这等价于：

$$y_i\left(\sum_{j=1}^{\ell} y_j\hat{\alpha}_j\left(\mathbf{x}_j\cdot\mathbf{x}_i\right)+b\right)\geqslant 1$$

然而它不能保证一定成立，所以在最大间隔情况下可行间隙不能直接计算得出。从下面可以看到，在某种软间隔情况下，可行间隙是可以估计的。

仅有拉格朗日乘子的某个子集是非零的事实，称为稀疏性，这意味着支持向量包括了重构超平面的所有必要信息。即使移除所有的其他点，仍然可以为剩余的支持向量子集找到相同的最大间隔超平面。这可以从对偶问题中看出，去除非支持向量的行和列，对剩余的子矩阵仍有相同的优化问题。因此，最优解保持不变。根据第 4.4 节的定义，最大间隔超平面是一个压缩方案，既然给定了支持向量的子集，可以重构能正确分类整个训练集的最大间隔超平面。应用定理 4.25，可以得到下面的结论。

定理 6.8 *考虑在内积空间 X 上具有单位权重向量的阈值化实值线性函数 $\mathscr{L}$。对 $X\times\{-1,1\}$ 上任意概率分布 $\mathscr{D}$，最大间隔超平面在 ℓ 个随机样例 S 上的误差以概率 $1-\delta$ 不大于：*

$$\underset{\mathscr{D}}{\mathrm{err}}(f)\leqslant\frac{1}{\ell-d}\left(d\log\frac{e\ell}{d}+\log\frac{\ell}{\delta}\right)$$

这里 $d=\#\mathrm{sv}$ 是支持向量的数目。

该定理显示支持向量的数目越少，泛化能力越强。这与寻找分类函数的紧凑表示的 Ockham 准则有密切联系。界的良好性能在于它不是与特征空间的维数显式相关。

图 6.2 显示了具有不同 σ 值的高斯核函数的支持向量机从一致分布的点中学习棋盘得到的结果。白点代表正点，黑点代表负点。支持向量用大的点表示。浅色区域包含了决策函数分类为正的点，而深色区域包含分类为负的点。注意两种情况下训

评注 6.9 所期望的泛化误差的一个稍严格的界可以用相同量表示，并用留一法获得。当一个非支持向量被忽略时，它可以由训练数据的剩余子集正确分类，泛化误差的留一法估计是：

$$\frac{\#\,\mathrm{sv}}{\ell}$$

训练子集的循环置换显示测试点的期望误差可以用这个量给出界，但是使用的期望泛化界不保证它的方差，也就是不保证可靠性。事实上，留一法的界也受限于这个问题。定理 6.8 可看成在最大间隔分类器的情况下一个仅稍弱的界以高的概率成立，并且在这种情况下方差不会太高。

最大间隔分类器没有试图控制支持向量的数目，但实践中通常只有很少的支持向量。解的稀疏性也促使产生了很多实现技术来处理大的数据集，这将在第 7 章深入讨论。

最大间隔算法仅有的自由度是核的选择，它需要模型选择。对问题的所有先验知识都可以帮助选择参数化的核，模型选择转化为调整参数的问题。对多数类型的核函数，比如多项式或者高斯核函数，总有可能找到一个核参数使得数据是可分的。但是，一般地说，强迫分开数据容易导致过拟合，尤其是数据中有噪声的时候。

在这种情况下，离群点的拉格朗日乘子通常很大，因此训练数据可以根据正确分开的难易度排序，从而可以用于数据筛选。

该算法在这个主题上提供了一个起点，过去的几年中在这个基础上提出了许多改进，它们企图解决该算法的几个弱点：对于噪声敏感，仅考虑两类，没有显式设计可得到稀疏解。

评注 6.10 注意在 SVM 中间隔有两方面作用。一是它的最大化确保了低的宽打散维，因此有较好的泛化性；二是不等式约束产生了 Karush-Kuhn-Tucker 互补条件，间隔产生了解向量的稀疏性。

6.1.2 软间隔优化

最大间隔分类器是一个重要概念，是分析和构造更加复杂的支持向量机的起点，但它在许多现实世界的问题中不能使用（第 8 章将给出一个特例）：如果数据有噪声，特征空间一般不能线性分开（除非使用很强的核，但很强的核易导致过拟合）。最大间隔分类器的主要问题是它总是完美地产生一个没有训练误差的一致假设。这当然是间隔度量的界促使产生的结果，间隔这个量当数据不能完全分开时它是负数。

依赖于像间隔这样的量使得系统易落入少数点所控制的危险境况。在真实数据中，噪声总是存在的，这会导致算法出现问题。进一步说，数据在特征空间中都不

能线性分开的情况下，原问题的可行区域是空的而对偶问题是无界的目标函数，这样优化问题不能得到解决。这些问题促使使用第 4 章介绍的间隔分布这类更稳健的度量。这样的度量能够容忍噪声和离群点，并能顾及更多的训练点，而不只是靠近边界的那些点。

回顾最大间隔情况下的原始优化问题如下：

$$\begin{array}{ll}\text{minimise}_{\mathbf{w},b} & (\mathbf{w}\cdot\mathbf{w}) \\ \text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geqslant 1 \quad i=1,\ldots,\ell\end{array}$$

为了能优化间隔松弛因子，需要引入松弛变量，它允许在一定程度上违反间隔约束：

$$\begin{array}{ll}\text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \quad i=1,\ldots,\ell \\ & \xi_i \geqslant 0 \;\; i=1,\ldots,\ell\end{array}$$

第 4 章的定理 4.22 用间隔松弛向量的二阶范数给出了泛化误差界。所谓二阶范数软间隔，包括权重向量 w的范数尺度化的 ξ_i。因此，泛化性所依赖的等价表达式是：

$$\begin{aligned}\frac{R^2+\frac{\|\xi\|_2^2}{\|\mathbf{w}\|_2^2}}{\gamma^2} &= \|\mathbf{w}\|_2^2\left(R^2+\frac{\|\xi\|_2^2}{\|\mathbf{w}\|_2^2}\right) \\ &= \|\mathbf{w}\|_2^2 R^2+\|\xi\|_2^2\end{aligned}$$

它指出在所得优化问题的目标函数中 C 的一个最优选择应该是 R^{-2}：

$$\begin{array}{ll}\text{minimise}_{\xi,\mathbf{w},b} & \langle \mathbf{w}\cdot\mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i^2 \\ \text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \quad i=1,\ldots,\ell \\ & \xi_i \geqslant 0 \;\; i=1,\ldots,\ell\end{array} \tag{6.3}$$

注意如果 $\xi_i<0$，则令 $\xi_i=0$，第一个约束仍然保持，这个变化将减小目标函数的值。通过去除 ξ_i 上的正约束获得的最优解与方程（6.3）的解是一致的。因此可以通过求解下面的优化问题得到方程（6.3）的解：

$$\begin{array}{ll}\text{minimise}_{\xi,\mathbf{w},b} & \langle \mathbf{w}\cdot\mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i^2 \\ \text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \quad i=1,\ldots,\ell\end{array} \tag{6.4}$$

实践中，参数 C 是一个变化范围大的值，优化性能的评价是通过使用独立的验证集或一种称为交叉验证的技术，后者仅在一个训练集上验证性能。参数 C 在一定范围内变化，$\|\mathbf{w}\|_2$ 会有相应的连续变化。因此，对特定的问题，选择 C 的值对应着选择 $\|\mathbf{w}\|_2$ 的值，然后在 w下最小化 $\|\xi\|_2$。这个思路在下面的一阶情况中采用，它最小化权重向量的范数和松弛变量一阶范数的组合，不完全符合定理 4.24：

$$\begin{array}{ll} \text{minimise}_{\xi,w,b} & \langle \mathbf{w}\cdot\mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \quad i=1,\ldots,\ell \\ & \xi_i \geqslant 0 \;\; i=1,\ldots,\ell \end{array} \tag{6.5}$$

既然有一个 C 的值对应着 $\|\mathbf{w}\|_2$ 的最优选择，这个 C 值也给出最优界，这个界对应给定 $\|\mathbf{w}\|_2$ 下找到的 $\|\xi\|_1$ 的最小值。

下面两个小节深入研究两个间隔松弛向量问题的对偶形式，也就是所谓软间隔算法。

二阶范数软间隔——对角权重

方程（6.4）下问题的原拉格朗日函数是：

$$L(\mathbf{w},b,\xi,\alpha) = \frac{1}{2}\langle \mathbf{w}\cdot\mathbf{w}\rangle + \frac{C}{2}\sum_{i=1}^{\ell}\xi_i^2 - \sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) - 1 + \xi_i\right]$$

这里 $\alpha_i \geqslant 0$ 是拉格朗日乘子，这在第 5 章描述过。

相应的对偶形式可以通过对应 $\mathbf{w},\xi,b$ 求偏导，置零：

$$\begin{aligned} \frac{\partial L(\mathbf{w},b,\xi,\alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} y_i\alpha_i\mathbf{x}_i = 0 \\ \frac{\partial L(\mathbf{w},b,\xi,\alpha)}{\partial \xi} &= C\xi - \alpha = 0 \\ \frac{\partial L(\mathbf{w},b,\xi,\alpha)}{\partial b} &= \sum_{i=1}^{\ell} y_i\alpha_i = 0 \end{aligned}$$

将得到的等式代入原拉格朗日函数，可以得到对偶目标函数下面的修正：

$$\begin{aligned} L(\mathbf{w},b,\xi,\alpha) &= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_iy_j\alpha_i\alpha_j\langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle + \frac{1}{2C}\langle \alpha\cdot\alpha\rangle - \frac{1}{C}\langle \alpha\cdot\alpha\rangle \\ &= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_iy_j\alpha_i\alpha_j\langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle - \frac{1}{2C}\langle \alpha\cdot\alpha\rangle \end{aligned}$$

因此，在 α 上最大化上述目标函数等价于最大化：

$$W(\alpha) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_iy_j\alpha_i\alpha_j\left(\langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}\right)$$

这里 δ_{ij} 是 Kronecker δ，当 $i=j$ 时定义为 1，其余为 0。对应的 Karush-Kuhn-Tucker 互补条件是：

$$\alpha_i \left[y_i((\mathbf{x}_i \cdot \mathbf{w}) + b) - 1 + \xi_i \right] = 0 \quad i = 1, \ldots, \ell$$

下面是推广到更一般的核版本的一些结果。

命题 6.11　考虑分类一个训练样本：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

在核 $K(\mathbf{x}, \mathbf{z})$ 隐式定义的特征空间中，假定参数 α^* 是下面的二次优化问题的解：

$$\begin{array}{ll} \textit{maximise} & W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \left(K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C} \delta_{ij} \right) \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0 \\ & \alpha_i \geqslant 0 \quad i = 1, \ldots, \ell \end{array}$$

令 $f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$，这里选择 b^* 使得 $y_i f(\mathbf{x}_i) = 1 - \alpha_i^* / C$ 成立，其中对任意 i 有 $\alpha_i^* \neq 0$。决策规则由 $\mathrm{sgn}(f(\mathbf{x}))$ 给出，这里等价于核 $K(\mathbf{x}, \mathbf{z})$ 隐式定义的特征空间中的最大间隔超平面，它是优化问题（6.3）的解，其中松弛变量的定义与几何间隔相关：

$$\gamma = \left(\sum_{i \in sv} \alpha_i^* - \frac{1}{C} \langle \alpha^* \cdot \alpha^* \rangle \right)^{-1/2}$$

证明　使用关系 $\alpha_i = C\xi_i$ 选择 b^* 的值，通过 Karush-Kuhn-Tucker 互补条件：

$$\alpha_i \left[y_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b \right) - 1 + \xi_i \right] = 0 \quad i = 1, \ldots, \ell$$

得到原始约束在非零 α_i 下一定也是等式。下面需要计算 $\mathbf{w}^*$ 的范数，它定义了几何间隔的大小：

$$\begin{aligned} \langle \mathbf{w}^* \cdot \mathbf{w}^* \rangle &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{j \in sv} \alpha_j^* y_j \sum_{i \in sv} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{j \in sv} \alpha_j^* \left(1 - \xi_j^* - y_j b^* \right) \\ &= \sum_{i \in sv} \alpha_i^* - \sum_{i \in sv} \alpha_i^* \xi_i^* \\ &= \sum_{i \in sv} \alpha_i^* - \frac{1}{C} \langle \alpha^* \cdot \alpha^* \rangle \end{aligned}$$

这仍然是一个二次规划问题，可以用与解最大间隔超平面相同的方法解。仅有的变化是增加了一个因子 $1/C$ 到训练集关联的内积矩阵的对角项上。产生的影响是在

矩阵的特征值上增加了一个因子 $1/C$，使得问题有更好的条件。因此可以将这个问题简单视做核的一个变化：

$$K'(\mathbf{x},\mathbf{z}) = K(\mathbf{x},\mathbf{z}) + \frac{1}{C}\delta_{\mathbf{x}}(\mathbf{z})$$

一阶范数软间隔——盒约束

一阶范数软间隔优化问题对应的拉格朗日函数是：

$$\begin{aligned} L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r}) &= \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C\sum_{i=1}^{\ell}\xi_i \\ &\quad -\sum_{i=1}^{\ell}\alpha_i\left[y_i((\mathbf{x}_i\cdot\mathbf{w})+b)-1+\xi_i\right] - \sum_{i=1}^{\ell}r_i\xi_i \end{aligned}$$

这里 $\alpha_i \geqslant 0, r_i \geqslant 0$。对偶表示可以通过求对应于 $\mathbf{w},\boldsymbol{\xi},b$ 的偏导，置零得到：

$$\begin{aligned} \frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} y_i\alpha_i\mathbf{x}_i = \mathbf{0} \\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r})}{\partial \xi_i} &= C-\alpha_i-r_i = 0 \\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r})}{\partial b} &= \sum_{i=1}^{\ell} y_i\alpha_i = 0 \end{aligned}$$

将上面的等式代入原拉格朗日函数得到对偶目标函数下面的修正：

$$L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r}) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_iy_j\alpha_i\alpha_j(\mathbf{x}_i\cdot\mathbf{x}_j)$$

很有意思的是，它与最大间隔的目标函数相同。仅有的区别是约束 $C-\alpha_i-r_i=0$ 和 $r_i \geqslant 0$，它实现了 $\alpha_i \leqslant C$，当 $\xi_i \neq 0$，仅当 $r_i = 0$ 时，有 $\alpha_i = C$。Karush-Kuhn-Tucher 互补条件成为：

$$\begin{aligned} &\alpha_i\left[y_i((\mathbf{x}_i\cdot\mathbf{w})+b)-1+\xi_i\right] = 0 \quad && i = 1,\ldots,\ell \\ &\xi_i(\alpha_i - C) = 0 && i = 1,\ldots,\ell \end{aligned}$$

注意，Karush-Kuhn-Tucker 条件意味着仅当 $\alpha_i = C$ 时出现非零的松弛变量。非零松弛变量的点有 $1/\|\mathbf{w}\|$ 间隔误差，它们的几何间隔小于 $1/\|\mathbf{w}\|$。而 $0 < \alpha_i < C$ 的点位于超平面 $1/\|\mathbf{w}\|$ 的距离处。因此有下面的命题。

命题 6.12 考虑分类一个训练样本：

$$S = ((\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_\ell, y_\ell))$$

在核$K(\mathbf{x},\mathbf{z})$隐式定义的特征空间中，假定参数$\boldsymbol{\alpha}^*$是下面的二次优化问题的解：

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i\alpha_i = 0 \\ & C \geqslant \alpha_i \geqslant 0 \quad i = 1,\ldots,\ell \end{array} \tag{6.6}$$

令$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i\alpha_i^* K(\mathbf{x}_i,\mathbf{x}) + b^*$，这里选择$b^*$使得$y_i f(\mathbf{x}_i) = 1$成立，其中对任意 i 有$C > \alpha_i^* > 0$。决策规则由$\operatorname{sgn}(f(\mathbf{x}))$给出，它等价于解决优化问题（6.5）的核$K(\mathbf{x},\mathbf{z})$隐式定义的特征空间中的超平面，这里松弛变量的定义与几何间隔相关：

$$\gamma = \left(\sum_{i,j\in sv} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i,\mathbf{x}_j)\right)^{-1/2}$$

证明　使用 Karush-Kuhn-Tucker 互补条件选择b^*的值，这个条件意味着如果$C > \alpha_i^* > 0$，而$\xi_i^* = 0$，并且：

$$y_i(\langle \mathbf{x}_i \cdot \mathbf{w}^*\rangle + b^*) - 1 + \xi_i^* = 0$$

$\mathbf{w}^*$的范数明显可以由下式给出：

$$\begin{aligned} \langle \mathbf{w}^* \cdot \mathbf{w}^*\rangle &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i,\mathbf{x}_j) \\ &= \sum_{j\in sv}\sum_{i\in sv} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i,\mathbf{x}_j) \end{aligned}$$

令人惊讶的是这个问题等价于带有附加约束的最大间隔超平面的问题，其中约束是对所有的α_i以 C 为上界。通常用盒约束称呼这个方程，因为向量$\boldsymbol{\alpha}$被约束到边长为 C 的盒子里。精度和正则化的妥协参数直接控制了α_i的大小。这直接意味着盒约束限制了离群点的影响，而离群点的拉格朗日乘子通常很大。约束也确保了可行区域的界，因此原问题总有非空可行区域。

评注 6.13　软间隔技术的一个问题是参数 C 的选择。典型的方法是在一个范围内试验，直到找到对特定训练集最好的选择。特征空间进一步也会影响参数的尺度。对优化问题（6.6），不同的 C 值得到的解和下面的优化问题中 ν 从 0 到 1 变化得到的解相同：

$$\begin{array}{ll} \text{maximise} & W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \\ \text{subject to} & \sum_{i=1}^{\ell} y_i\alpha_i = 0 \\ & \sum_{i=1}^{\ell} \alpha_i \geqslant \nu \\ & 1/\ell \geqslant \alpha_i \geqslant 0 \quad i = 1,\ldots,\ell \end{array}$$

在这个参数化过程中，ν给出了α_i加和的下界，它从目标函数中去除了线性项。可以

看出v是间隔误差的训练集的比例的上界，同时v又是支持向量全部数目与全部样例数的比例的下界。因此，v给出了问题的一个更透明的参数，它与特征空间的尺度无关，而仅与数据的噪声程度有关。这个方法的细节及其在回归中的应用，可以参考第6.5节提供的链接。

在一阶范数间隔松弛向量优化的情况下，可以计算可行间隙，因为在对偶形式下没有确定ξ_i，它可以通过下面的公式选择，这样确保原问题可解：

$$\xi_i = \max\left(0, 1 - y_i\left(\sum_{j=1}^{\ell} y_j\alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b\right)\right)$$

这里$\boldsymbol{\alpha}$是对偶问题的当前估计，选择b对某些i和$C > \alpha_i > 0$有$y_i f(\mathbf{x}_i) = 1$。一旦原问题可解，原目标和对偶目标的值之间的间距成为Karush-Kuhn-Tucker互补条件的和，这需要构造拉格朗日函数：

$$\begin{aligned}-L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r}) + \frac{1}{2}\langle \mathbf{w}\cdot\mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i = \\ = \sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\sum_{j=1}^{\ell} y_j\alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b\right) - 1 + \xi_i\right] + \sum_{i=1}^{\ell} r_i\xi_i\end{aligned}$$

这里$r_i = C - \alpha_i$。因此，使用$\boldsymbol{\alpha}$上的约束，原目标和对偶目标的间隙由下式给出：

$$\begin{aligned}\sum_{i=1}^{\ell}\alpha_i\left[y_i(\langle \mathbf{x}_i\cdot\mathbf{w}\rangle + b) - 1 + \xi_i\right] + \sum_{i=1}^{\ell} r_i\xi_i = \\ = \sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\sum_{j=1}^{\ell} y_j\alpha_j K(\mathbf{x}_j, \mathbf{x}_i)\right) - 1\right] + C\sum_{i=1}^{\ell}\xi_i \\ = \sum_{i,j=1}^{\ell}\alpha_i y_i y_j\alpha_j K(\mathbf{x}_j, \mathbf{x}_i) - \sum_{i=1}^{\ell}\alpha_i + C\sum_{i=1}^{\ell}\xi_i \\ = \sum_{i=1}^{\ell}\alpha_i - 2W(\boldsymbol{\alpha}) + C\sum_{i=1}^{\ell}\xi_i\end{aligned}$$

评注 6.14 这解释了为什么要强调在更复杂版本的学习器中最大（或硬）间隔是一个重要的概念：一阶和二阶软间隔学习器都产生了与最大间隔学习器相关的优化问题。

评注 6.15 从历史角度看，先引入了软间隔学习器，然后用间隔分布泛化界为项做了修正。因此一阶范数更接近百分误差界，更能被接受。结果显示泛化性的一阶和二阶界都存在。哪种方法在实践中表现更好将取决于数据，也可能会受到噪声类型

6.1.3 线性规划支持向量机

如果不使用间隔分布上的泛化界，还可以尝试强化其他学习偏置，比如定理 4.25 和定理 6.8 给出的样本压缩界。这导向了寻找最稀疏分开超平面这样的算法，而不考虑它的间隔。这个问题计算量很大，但可以通过最小化正乘子数目的估计 $\sum_{i=1}^{\ell}\alpha_i$ 来近似解决，同时使间隔为 1。松弛变量的引入与上面给出的类似，可以直接用子对偶表示中，得到下面的线性优化问题：

$$\begin{array}{ll}\text{minimise} & L(\boldsymbol{\alpha},\boldsymbol{\xi}) = \sum_{i=1}^{\ell}\alpha_i + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to} & y_i\left[\sum_{j=1}^{\ell}\alpha_i\langle \mathbf{x}_i,\mathbf{x}_j\rangle + b\right] \geqslant 1-\xi_i \quad i=1,\ldots,\ell \\ & \alpha_i \geqslant 0, \xi_i \geqslant 0 \quad i=1,\ldots,\ell\end{array}$$

这种类型方法的开发与隐含在标准 SVM 定义中的二阶范数最大间隔无关。它的优点是求解一个线性规划问题，而不是 SVM 中的凸二次规划问题。这个算法中也可以应用核来得到隐式特征空间。泛化性界直接与 $\sum_{i=1}^{\ell}\alpha_i$ 相关，这是近期得出的结论。

6.2 支持向量回归

支持向量的方法也可以应用到回归问题中，仍保留了最大间隔算法的所有主要特征：非线性函数可以通过核特征空间中的线性学习器得到，同时系统的容量由与特征空间维数不相关的参数控制。同分类算法一样，学习算法要最小化一个凸函数，并且它的解是稀疏的。

同分类算法的思路一样，这里的算法也需要优化第 4 章的回归泛化界。这需要定义一个损失函数，它可以忽略真实值某个上下范围内的误差。这种类型的函数也就是 ε 不敏感损失函数。既然术语这么标准，尽管前面曾经为泛化误差，也就是随机测试样例的误分概率保留了 ε 这个符号，本书还是冒险使用它作为损失的表示。

图 6.4 显示了具有 ε 不敏感带的一维线性回归函数的例子。变量 ξ 度量了训练点上误差的代价。在 ε 不敏感区内的点误差为 0。图 6.5 显示了非线性回归函数的类似情况。

损失函数有许多合理的选择，它的解以函数的最小化为特征。探讨 ε 不敏感损失函数的另一个动机是如同分类 SVM 一样，它可以确保对偶变量的稀疏性。使用训练点的一个小的子集来表示解有很大的计算优势。使用 ε 不敏感损失函数就有这个优势，同时确保全局最小解的存在和可靠泛化界的优化。

本节首先描述 ε 不敏感损失，然后从第 4 章的界得出两个算法，分别与损失向量的一阶和二阶范数有关。为了比较，将给出特征空间的岭回归算法，它不具有稀疏性，所以有更多的实现问题。最后给出了基于高斯过程的一个流行的回归算法，并

讨论它是如何等价于特征空间中的岭回归算法，因此直接与 SVM 密切相关。

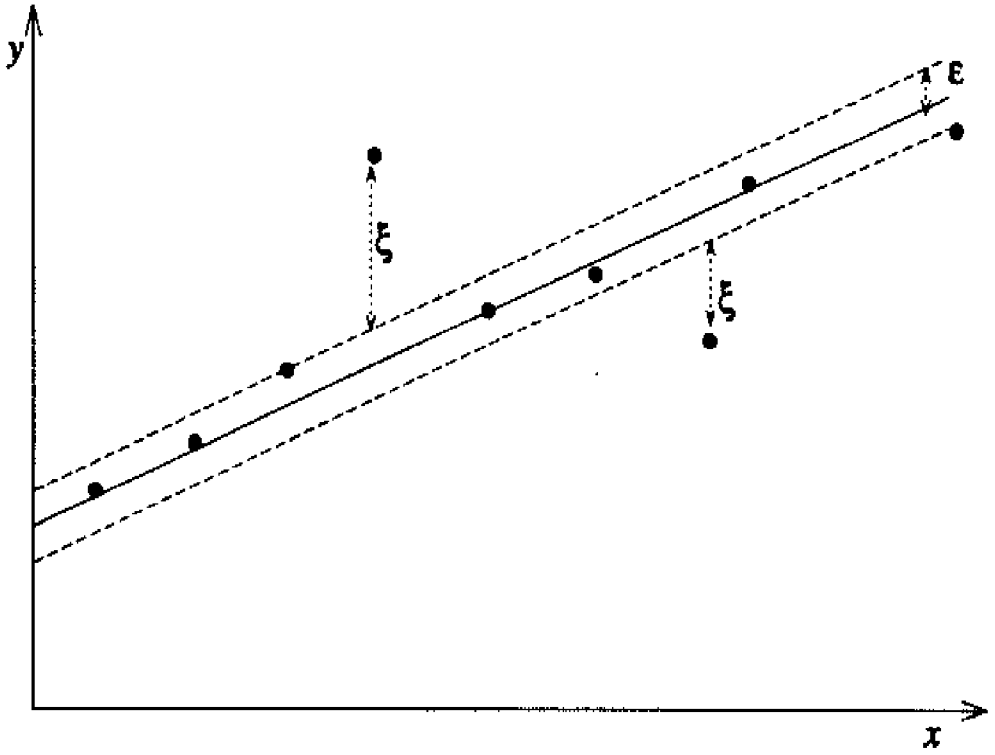

图 6.4　一维线性回归问题的不敏感带

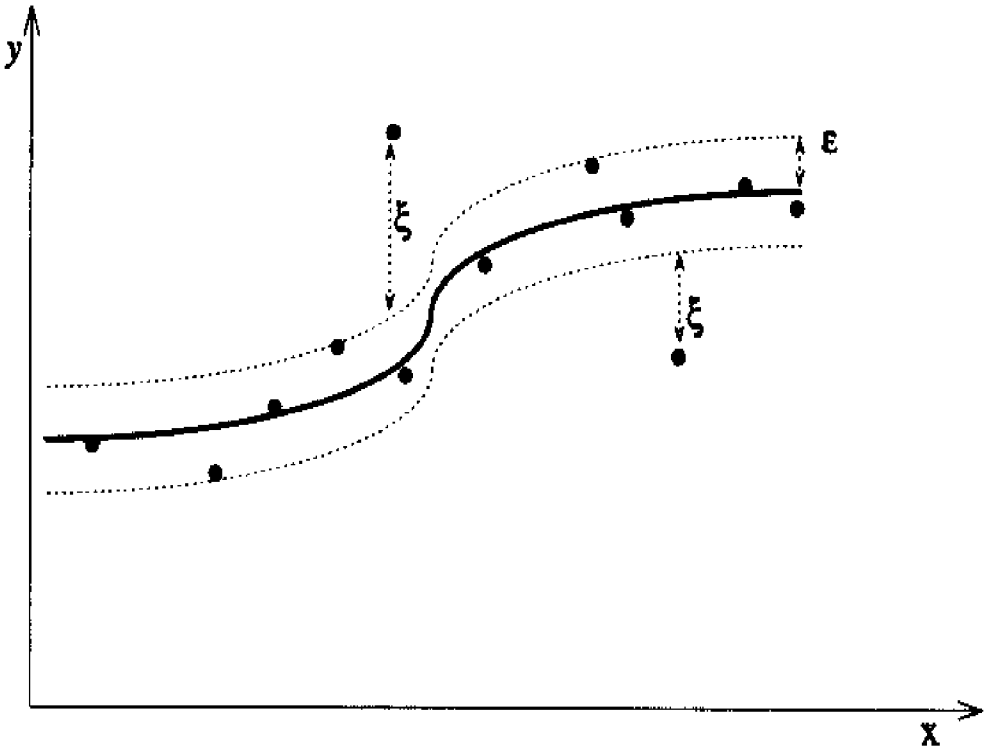

图 6.5　非线性回归函数的不敏感带

6.2.1 ε不敏感损失回归

定理 4.28 和定理 4.30 给出了线性回归器的泛化性界，它以权重向量的范数和松弛变量的二阶和一阶范数表示。ε不敏感损失函数等价于这些松弛变量。

定义 6.17 （线性）ε不敏感损失函数 $L^{\varepsilon}(\mathbf{x}, y, f)$ 定义为：

$$L^{\varepsilon}(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_{\varepsilon} = \max\left(0, |y - f(\mathbf{x})| - \varepsilon\right)$$

这里 f 是域 X 上的实值函数，$\mathbf{x} \in \mathbf{X}$ 并且 $y \in \mathbb{R}$。类似地，二次ε不敏感损失由下式给出：

$$L_2^{\varepsilon}(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_{\varepsilon}^2$$

如果将这个损失函数同定义 4.27 中的间隔松弛向量做比较，可以直接发现间隔松弛变量 $\xi((\mathbf{x}_i, y_i), f, \theta, \gamma)$ 满足：

$$\xi((\mathbf{x}_i, y_i), f, \theta, \gamma) = L^{\theta-\gamma}(\mathbf{x}_i, y_i, f)$$

因此，如上所示，第 4 章的结果使用了ε不敏感损失函数，其中 $\varepsilon = \theta - \gamma$。图 6.6 和图 6.7 显示函数 $y - f(\mathbf{x})$ 在ε为 0 和非 0 时，线性和二次ε不敏感损失的形式。

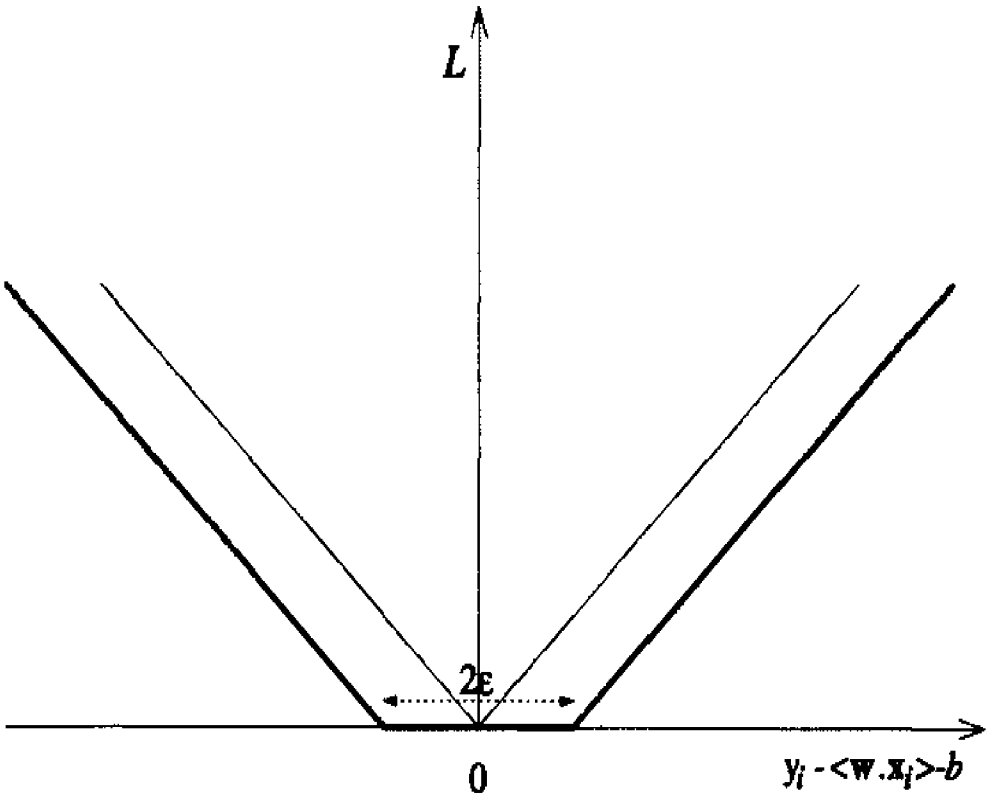

图 6.6 ε为 0 或非 0 所对应的线性ε不敏感损失

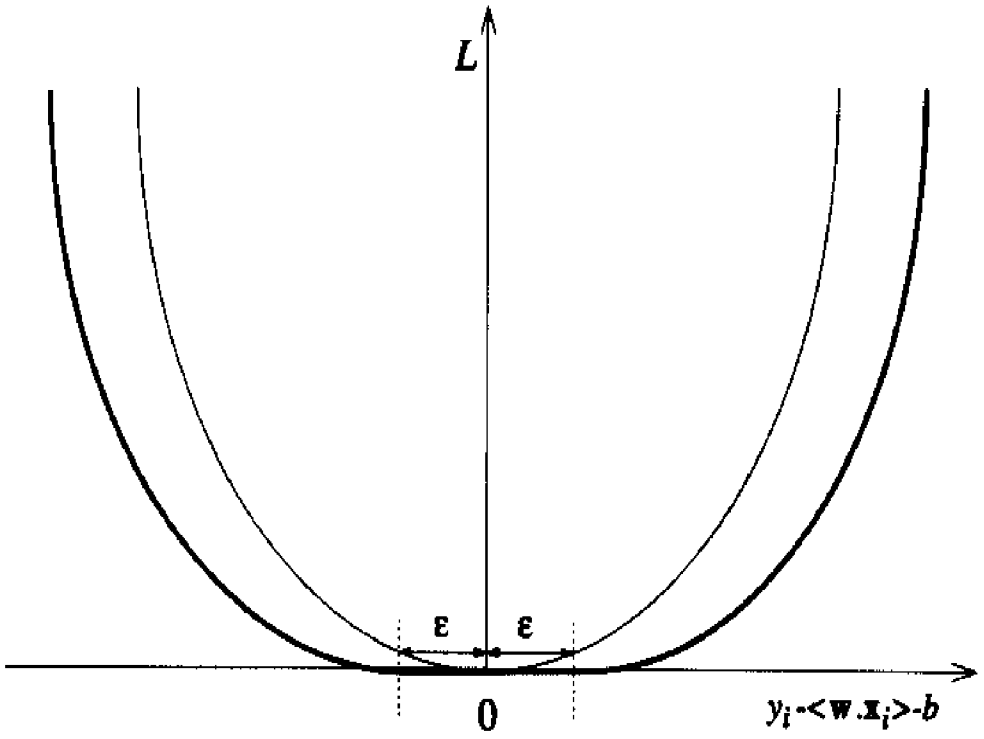

图 6.7　ε为 0 或非 0 所对应的二次ε不敏感损失

二次ε不敏感损失

定理 4.28 建议通过最小化二次ε不敏感损失的和：

$$R^2\|\mathbf{w}\|^2+\sum_{i=1}^{\ell}L_2^{\varepsilon}(\mathbf{x}_i,y_i,f)$$

来优化回归器的泛化性。这里f是权重向量w定义的函数。最小化这个量的优势是在所有γ上最小化界，这意味着它在所有$\theta=\varepsilon+\gamma$的值上最小化。如同分类情况下，这里引入参数$C$来度量复杂性和损失的妥协。因此原问题定义如下：

$$\begin{array}{ll}\text{minimise} & \|\mathbf{w}\|^2+C\sum_{i=1}^{\ell}(\xi_i^2+\hat{\xi}_i^2)\\ \text{subject to} & (\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b)-y_i\leqslant\varepsilon+\xi_i \quad i=1,\ldots,\ell\\ & y_i-(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle+b)\leqslant\varepsilon+\hat{\xi}_i \quad i=1,\ldots,\ell\\ & \xi_i,\hat{\xi}_i\geqslant 0 \quad i=1,\ldots,\ell\end{array} \tag{6.7}$$

这里引入了两个松弛变量，一个是在目标值之上超出ε所设，另一个是在目标值之下超出ε所设。考虑为变化的C值求解这个方程，然后使用验证的方法选择参数的最优值。可以使用标准方法导出对偶问题，考虑$\xi_i\hat{\xi}_i=0$，和类似的关系$\alpha_i\hat{\alpha}_i=0$，对下面的拉格朗日乘子成立：

$$\begin{array}{ll}\text{maximise} & \sum_{i=1}^{\ell} y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{\ell}(\hat{\alpha}_i + \alpha_i) \\ & \quad -\frac{1}{2}\sum_{i,j=1}^{\ell}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_i - \alpha_j)\left(\langle \mathbf{x}_i \cdot \mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}\right) \\ \text{subject to} & \sum_{i=1}^{\ell}(\hat{\alpha}_i - \alpha_i) = 0 \\ & \hat{\alpha}_i \geqslant 0,\ \alpha_i \geqslant 0 \qquad i = 1,\ldots,\ell\end{array}$$

它对应的 Karush-Kuhn-Tucker 互补条件是：

$$\begin{array}{ll}\alpha_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b - y_i - \varepsilon - \xi_i\right) = 0 & i = 1,\ldots,\ell \\ \hat{\alpha}_i\left(y_i - \langle \mathbf{w}\cdot\mathbf{x}_i\rangle - b - \varepsilon - \hat{\xi}_i\right) = 0 & i = 1,\ldots,\ell \\ \xi_i\hat{\xi}_i = 0,\ \alpha_i\hat{\alpha}_i = 0 & i = 1,\ldots,\ell\end{array}$$

评注 6.18 注意通过替代 $\boldsymbol{\beta} = \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}$ 并且使用关系 $\alpha_i\hat{\alpha}_i = 0$，可以重写对偶问题，使其更接近分类情况下的形式：

$$\begin{array}{ll}\text{maximise} & \sum_{i=1}^{\ell} y_i\beta_i - \varepsilon\sum_{i=1}^{\ell}|\beta_i| - \frac{1}{2}\sum_{i,j=1}^{\ell}\beta_i\beta_j\left(\langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}\right) \\ \text{subject to} & \sum_{i=1}^{\ell}\beta_i = 0 \qquad i = 1,\ldots,\ell\end{array}$$

对 $y_i \in \{-1,1\}$，当 $\varepsilon = 0$ 时相似性更加明显，如果使用变量 $\hat{\beta}_i = y_i\beta_i$，仅有差别是没有限制 $\hat{\beta}_i$ 为正值，而分类情况下 α_i 是正值。事实上，此后使用这种形式时，是用 $\boldsymbol{\alpha}$ 来替代 $\boldsymbol{\beta}$ 的。

评注 6.19 ε 非 0 产生的影响是引入了包含对偶参数的额外权重衰减因子。$\varepsilon = 0$ 的情况对应的是带有由参数 C 控制的权重衰减因子的标准最小二乘线性回归。而当 $C \to \infty$，问题趋向于无约束的最小二乘，这等价于保持内积矩阵对角不变。注意，在本章最后给出对更一般的损失函数的研究。

下面是更一般的核版本的结果。

命题 6.20 假定在训练集：

$$S = ((\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_\ell, y_\ell))$$

上使用核 $K(\mathbf{x},\mathbf{z})$ 隐式定义的特征空间做回归，并且假定参数 $\boldsymbol{\alpha}^*$ 解二次优化问题：

$$\begin{array}{ll}\textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} y_i\alpha_i - \varepsilon\sum_{i=1}^{\ell}|\alpha_i| \\ & \quad -\frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j\left(K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{C}\delta_{ij}\right) \\ \textit{subject to} & \sum_{i=1}^{\ell}\alpha_i = 0\end{array}$$

令 $f(\mathbf{x}) = \sum_{i=1}^{\ell}\alpha_i^* K(\mathbf{x}_i,\mathbf{x}) + b^*$，这里选择 b^* 使得 $f(\mathbf{x}_i) - y_i = -\varepsilon - \alpha_i^*/C$ 对任意 i 在 $\alpha_i^* > 0$ 下成立。则函数 $f(\mathbf{x})$ 等价于在核 $K(\mathbf{x},\mathbf{z})$ 隐式定义的特征空间中求解优化问题（6.7）得到的超平面。

线性 ε 不敏感损失

定理 4.30 要求对参数 C 的某些值最小化线性 ε 不敏感损失的和：

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell} L^{\varepsilon}(\mathbf{x}_i, y_i, f)$$

如同在分类情况下对固定训练集，参数 C 可以控制 $\|\mathbf{w}\|$ 的大小。等价的原问题定义如下：

$$\begin{array}{ll} \text{minimise} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \hat{\xi}_i) \\ \text{subject to} & (\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) - y_i \leqslant \varepsilon + \xi_i \\ & y_i - (\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) \leqslant \varepsilon + \hat{\xi}_i \\ & \xi_i, \hat{\xi}_i \geqslant 0 \quad i = 1, 2, \ldots, \ell \end{array} \tag{6.8}$$

相应的对偶问题可用标准方法导出：

$$\begin{array}{ll} \text{maximise} & \sum_{i=1}^{\ell}(\hat{\alpha}_i - \alpha_i)y_i - \varepsilon\sum_{i=1}^{\ell}(\hat{\alpha}_i + \alpha_i) \\ & \quad -\frac{1}{2}\sum_{i,j=1}^{\ell}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\langle \mathbf{x}_i \cdot \mathbf{x}_j\rangle \\ \text{subject to} & 0 \leqslant \alpha_i, \hat{\alpha}_i \leqslant C \quad i = 1, \ldots, \ell \\ & \sum_{i=1}^{\ell}(\hat{\alpha}_i - \alpha_i) = 0 \quad i = 1, \ldots, \ell \end{array}$$

它对应的 Karush-Kuhn-Tucker 互补条件是：

$$\begin{array}{ll} \alpha_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b - y_i - \varepsilon - \xi_i\right) = 0 & i = 1, \ldots, \ell \\ \hat{\alpha}_i\left(y_i - \langle \mathbf{w}\cdot\mathbf{x}_i\rangle - b - \varepsilon - \hat{\xi}_i\right) = 0 & i = 1, \ldots, \ell \\ \xi_i\hat{\xi}_i = 0, \alpha_i\hat{\alpha}_i = 0 & i = 1, \ldots, \ell \\ (\alpha_i - C)\xi_i = 0, (\hat{\alpha}_i - C)\hat{\xi}_i = 0 & i = 1, \ldots, \ell \end{array}$$

同评注 6.18 提到的一样，用 α_i 替代 $\hat{\alpha}_i - \alpha_i$，并且 $\alpha_i\hat{\alpha}_i = 0$，得到下面的命题。

命题 6.21　假定在训练集：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

上使用核 $K(\mathbf{x}, \mathbf{z})$ 隐式定义的特征空间做回归，并且假定参数 $\boldsymbol{\alpha}^*$ 是二次优化问题的解：

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} y_i\alpha_i - \varepsilon\sum_{i=1}^{\ell}|\alpha_i| - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \textit{subject to} & \sum_{i=1}^{\ell}\alpha_i = 0, -C \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, \ell \end{array}$$

令 $f(\mathbf{x}) = \sum_{i=1}^{\ell}\alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$，这里选择 b^* 使得 $f(\mathbf{x}_i) - y_i = -\varepsilon$ 对任意 i 在 $0 < \alpha_i^* < C$ 下成立。则函数 $f(\mathbf{x})$ 等价于核 $K(\mathbf{x}, \mathbf{z})$ 隐式定义的特征空间中求解问题（6.8）得到的超平面。

评注 6.22　考虑函数围绕学习算法所输出的 $\pm\varepsilon$ 带，不严格在管道内部的点支持向量。那些没有接触管道的点将有等于 C 的值。

评注 6.23　前面再次描述了最标准的优化方法。文献中包含了很多变化，包括探讨使用不同的范数和改变优化来控制 ε 带外点的数目。在这种情况下，是点的数目作为问题的输入，而不是 ε 的值。关于使用 SVM 做回归的这些进展将在第 6.5 节给出。

6.2.2 核岭回归

如同在评注 6.19 中提到的，二次损失函数中$\varepsilon = 0$对应着有权重衰减因子的最小二乘回归，又称为岭回归。下一小节会介绍它等价于高斯过程推导出的技术。因此本书将给出独立的推导，来阐明这些系统的联系。这些系统忽略了偏置项。（原）问题可以描述如下：

$$\begin{array}{ll}\text{minimise} & \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{\ell}\xi_i^2 \\ \text{subject to} & y_i - \langle \mathbf{w}\cdot\mathbf{x}_i\rangle = \xi_i \quad i = 1,\ldots,\ell\end{array} \tag{6.9}$$

从中可以得到拉格朗日函数：

$$\text{minimise}\quad L(\mathbf{w},\xi,\alpha) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{\ell}\xi_i^2 + \sum_{i=1}^{\ell}\alpha_i(y_i - \langle \mathbf{w}\cdot\mathbf{x}_i\rangle - \xi_i)$$

求导数置零，得到：

$$\mathbf{w} = \frac{1}{2\lambda}\sum_{i=1}^{\ell}\alpha_i\mathbf{x}_i \quad 和 \quad \xi_i = \frac{\alpha_i}{2}$$

重新替换这些关系，得到下面的对偶问题：

$$\text{maximise}\quad W(\alpha) = \sum_{i=1}^{\ell} y_i\alpha_i - \frac{1}{4\lambda}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j\langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle - \frac{1}{4}\sum\alpha_i^2$$

为了方便写成向量形式：

$$W(\alpha) = \mathbf{y}'\alpha - \frac{1}{4\lambda}\alpha'\mathbf{K}\alpha - \frac{1}{4}\alpha'\alpha$$

这里 $\mathbf{K}$ 表示 Gram 矩阵 $\mathbf{K}_{ij} = \langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle$，或者如果在核特征空间就是核矩阵 $\mathbf{K}_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)$。对应$\alpha$求导，置零得到下面的条件：

$$-\frac{1}{2\lambda}\mathbf{K}\alpha - \frac{1}{2}\alpha + \mathbf{y} = 0$$

得到解：

$$\alpha = 2\lambda(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$$

和相应的回归方程：

$$f(\mathbf{x}) = \langle \mathbf{w}\cdot\mathbf{x}\rangle = \mathbf{y}'(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}$$

这里 $\mathbf{k}$ 是项为 $k_i = \langle \mathbf{x}_i\cdot\mathbf{x}\rangle$, $i = 1,\ldots,\ell$的向量，因此，有下面的命题。

命题 6.24 假定在训练集：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

上使用核 $K(\mathbf{x}, \mathbf{z})$隐式定义的特征空间做回归，令 $f(\mathbf{x}) = \mathbf{y}'(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}$，这里 $\mathbf{K}$ 是项为 $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$的 $\ell \times \ell$ 的矩阵，$\mathbf{k}$ 是项为 $\mathbf{k}_i = K(\mathbf{x}_i, \mathbf{x})$的向量。则函数 $f(\mathbf{x})$ 等价于核 $K(\mathbf{x}, \mathbf{z})$隐式定义的特征空间中求解问题（6.9）得到的超平面。

这个算法有几个不同的名称。它也称为 Krieging，解称为正则化网络，其正则算子通过核隐式选择。下一小节使用高斯过程解贝叶斯学习问题时将得到相同的函数。

6.2.3　高斯过程

本小节将讨论第 4.6 节的贝叶斯学习和第 3.5 节的高斯过程。后验分布：

$$P(t, \mathbf{t}|\mathbf{x}, S) \propto P(\mathbf{y}|\mathbf{t})P(t, \mathbf{t}|\mathbf{x}, \mathbf{X})$$

其中 $\mathbf{y}$ 是训练集的输出值，假定已被噪声腐蚀；$\mathbf{t}$ 是与 $\mathbf{y}$ 相关的目标真实输出值。相关性如下分布：

$$P(\mathbf{y}|\mathbf{t}) \propto \exp\left[-\frac{1}{2}(\mathbf{y} - \mathbf{t})'\Omega^{-1}(\mathbf{y} - \mathbf{t})\right]$$

这里 $\Omega = \sigma^2\mathbf{I}$。第 3.5 节中介绍的高斯过程分布定义如下：

$$\begin{aligned} P(t, \mathbf{t}|\mathbf{x}, \mathbf{X}) &= P_{f\sim\mathcal{G}}[(f(\mathbf{x}), f(\mathbf{x}_1), \ldots, f(\mathbf{x}_\ell)) = (t, t_1, \ldots, t_\ell)] \\ &\propto \exp\left(-\frac{1}{2}\hat{\mathbf{t}}'\hat{\Sigma}^{-1}\hat{\mathbf{t}}\right) \end{aligned}$$

这里 $\hat{\mathbf{t}} = (t, t_1, \ldots, t_\ell)'$ 并且 $\hat{\Sigma}$ 行和列的索引都是从 0 到 ℓ。行从 1 到 ℓ 的主子矩阵是矩阵 Σ，这里对协方差函数 $K(\mathbf{x}, \mathbf{z})$有 $\Sigma_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$，同时项 $\hat{\Sigma}_{00} = K(\mathbf{x}, \mathbf{x})$，并且在 0 行和 0 列的项是：

$$\hat{\Sigma}_{0i} = \hat{\Sigma}_{i0} = K(\mathbf{x}, \mathbf{x}_i)$$

变量 t 的分布是预测分布。它是一个均值为 $f(\mathbf{x})$ 方差为 $V(\mathbf{x})$的高斯分布：

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{y}'(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k} \\ V(\mathbf{x}) &= K(\mathbf{x}, \mathbf{x}) - \mathbf{k}'(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k} \end{aligned} \tag{6.10}$$

这里 $\mathbf{K}$是项为 $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$的 $\ell \times \ell$ 的矩阵，$\mathbf{k}$ 是项为$\mathbf{k}_i = K(\mathbf{x}_i, \mathbf{x})$的向量。因此高斯过程估计的预测同命题 6.24 的岭回归函数非常一致，这里参数 λ 选择为噪声分布的方差。这强化了间隔松弛优化和数据中噪声的关系。它指出优化二阶范数[见问题（6.7）]对应于方差为 $\frac{1}{C}$ 的高斯噪声的假设。

高斯过程也以预测分布方差的形式对预测的可靠性做出估计。更重要的是这一分析可以用来估计支持协方差函数特定选择的证据。并且，可以通过最大化这个证据来适应性选择参数化核函数，因此是给定数据的最可能的选择。核函数或者协方差函数可以视做数据的模型，由此提供了一种模型选择的原理性方法。

6.3 讨论

本章包含了本书的核心材料。它显示如何用第 4 章的学习理论来避免线性函数在第 3 章中的高维核特征空间中应用的困难。并且显示了如何将分类和回归采用的方法对应的优化问题转化为凸对偶二次规划问题。在回归情况下，损失函数仅惩罚比阈值ε大的误差。这种损失函数通常产生决策规则的稀疏表示，从而具有重要的算法和表示优势。如果在优化间隔松弛向量的二阶范数情况下设置$\varepsilon=0$，可以使用一个对应协方差函数的高斯过程恢复回归器的输出，或者使用等价的岭回归函数。当$\varepsilon=0$时，这些方法的缺点是丧失了表示的稀疏性。

本书在所有算法的优化中所考虑的这种类型的条件也在其他上下文中出现，都得出了对偶表示的解。表示这些条件的一般方式是：

$$\|f\|_{\mathcal{H}}^2+C\frac{1}{\ell}\sum_{i=1}^{\ell}L(y_i,f(\mathbf{x}_i))$$

这里L是损失函数，$\|\cdot\|_{\mathcal{H}}$代表了正则化算子，C是正则化参数。如果L是平方损失函数，就产生了正则化网络，其中高斯过程是一个特殊情况。对这种类型的问题，解总是可以表示为对偶形式。

下一章将描述如何有效求解这些优化问题，当问题具有很大的数据集时，通常可以利用解的稀疏性。

6.4 习题

1. 当α是优化问题（6.2）的解，而γ是相应的几何间隔时下面四个表达式之间的关系是什么？$W(\alpha)$，$\sum_{i=1}^{\ell}\alpha_i$，$\sum_{i,j=1}^{\ell}y_iy_j\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_j)$和$\frac{1}{\gamma^2}$，如果数据不是线性可分的，优化问题会出现什么问题？这个问题在软间隔优化问题中是如何避免的？
2. 导出回归问题（6.7）的对偶表示形式，并证明命题 6.20。
3. 考虑下面的优化问题：

$$\begin{array}{ll}\text{minimise}_{\mathbf{w},b} & \langle \mathbf{w}\cdot\mathbf{w}\rangle + C_1\sum_{i=1}^{\ell}\xi_i + C_2\sum_{i=1}^{\ell}\xi_i^2 \\ \text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \quad i=1,\ldots,\ell \\ & \xi_i \geqslant 0 \quad i=1,\ldots,\ell\end{array}$$

讨论参数 C_1 和 C_2 变化产生的影响。导出对偶表示形式。

6.5 补充读物和高级主题

支持向量机是一类很特别的算法，特点是使用了核，没有局部最小，解的稀疏性，以及通过间隔或者是维数无关的量（比如支持向量个数）来控制容量。它们是由 Boser，Guyon 和 Vapnik[19]发明的，并首次在计算学习理论（COLT）1992 年的会议论文[19]中提出。算法中所有的这些特征已经存在，并已经从 20 世纪 60 年代开始在机器学习领域使用：很多人讨论过输入空间的大间隔超平面，比如 Duda 和 Hart [35]，Cover [28]，Vapnik 等人 [166,161]和几篇统计论文（比如[4]）。核的使用是由 Aronszajn[7]，Wahba[171] 和 Poggio[116]等提出的，1964 年 Aizermann 等人 [1] 的论文中提出了核在特征空间中作为内积的几何解释。Mangasarian[84]已经将类似的技术用于模式识别中，稀疏性也早已被讨论过[28]，相关的早期著作见[57]。使用松弛变量来解决噪声和不可分问题也在 20 世纪 60 年代由 Smith[143]介绍过，Bennett 和 Mangasarian[15]中做了改进。然而，直到 1992 年所有这些特征才集中到一起形成了最大间隔分类器，也就是基本的支持向量机。到 1995 年，引入软间隔版本[27]：令人惊奇的是所有这些部件如此自然和优雅的结合在一起。论文[138,10]给出了硬间隔 SVM 的第一个严格统计界，而论文[141]为软间隔算法和回归情况给出了类似的界。

在他们的引入之后，不断增多的研究者参与了这些系统的算法和理论分析，在短短几年内有效地创立了一个研究方向，融合了几个相差较大的领域和概念，比如统计、泛函分析、最优化和机器学习等。软间隔分类器是稍后几年由 Cortes 和 Vapnik[27]引入的，到 1995 年扩展到回归问题中[158]。

Vapnik 最近的两本著作[158,159]为这一领域提供了广泛的理论背景，并发展了支持向量机的概念。

既然核、学习理论和实现技术方面最新的进展在相关章节的最后一节给出，这里只是从整个算法的角度对一些改进做简要的回顾。[88]研究了算法的泛化性，[178,159,113]将其扩展到多类别情况下。一般的回归问题也已经研究过：Smola，Schölkopf 和 Mueller 讨论了一般的损失函数类[147]，[144,125]中讨论了特征空间的岭回归算法。

岭回归是正则化网络的特殊情况。正则化的概念是 Tikhonov[153]引入的，并由 Girosi 等人 [52] 应用到正则化网络的学习中。很多作者[51,171,172,146,38]研究过正

则化网络和支持向量机的关系。而正则化网络和神经网络的关系则早在 1990 年[116]就研究过，[52,39]中有完整的文献。

评注 6.13 中分类和回归的ν支持向量算法在[135]中引入，进一步的研究则在[131,130]中做了介绍。基本思路的改写已经应用到密度估计[167]、转导[13]、贝叶斯点估计[59]、序回归[60]等方面。Rifkin 等人[121]显示对某些退化的训练集，软间隔给出的是平凡解。

不包含在第 4 章框架中的理论进展有论文[34]给出的泛化性的分析，它提供了 SVM 的统计分析，论文[66,160,177,173,107]提供了期望误差的交叉验证分析，著作[159]以间隔和包含必要支持向量的最小球的半径给出了期望误差的界。

几位作者做过 SVM 概念的扩展，比如 Mangasarian 的推广 SVM[83]，特别有意思的是一个称为贝叶斯点学习器[59]的开发，它利用了贝叶斯泛化性理论的归纳原则。虽然丢失了稀疏性的特征，但系统展示了良好的性能，并例示了此类算法不局限于间隔界。类似 SVM 系统的另一个例子是高斯过程给出的，它不采用间隔界。

最近报告了 SVM 的大量实践，涉及很广泛的领域，比如生物信息学、计算语言学和计算机视觉（有些在第 8 章叙述）。多数进展收录在[132,149]，以及综述[23,145,39]中。多数新著作只能从因特网上得到，可以通过网站[30]。

最后，一些博士论文，比如 Cortes[26]、Schölkopf[129]和 Smola[148]提供了包括可行间隙工作在内的研究课题的有价值的第一手材料。

高斯过程的综述见[180]和 Rasmussen 的博士论文[120]。高斯过程扩展到分类情况的工作已经开始，但超出了本书的范围。

这些参考文献也在网站 www.support-vector.net 上给出，这个网站将不断及时补充新的研究成果并提供在线软件和论文的链接。

第7章 实现技术

上一章讨论了如何将支持向量机的训练转化为线性约束下的一个凸二次规划形式的问题。这样的凸二次规划没有局部最小并可有效求解。而且，这个问题的对偶表示显示了即使在高维特征空间中训练也很有效率。最小化许多变量的可微函数尤其是凸函数的问题已经过广泛研究，多数的标准方法都可以直接应用到SVM的训练中。然而，许多情况下，问题的特殊情况需要开发特定的技术。比如，实践中训练集很大是直接使用标准技术的障碍，因为仅仅存储核矩阵就需要一个随样本大小二次增长的内存空间，即使样本仅有几千个点，内存空间也要超出上百兆字节。

由此推动了支持向量机特定算法的设计，这些算法利用了解的稀疏性、优化问题的凸性和特征空间的隐式映射。所有这些特征有助于建立卓有效率的计算。解的优美的数学特性可以进一步为很大数据集提供停止条件和分解方法。

本章先简要回顾一些最通用的方法，然后深入描述一个特定的算法：序贯最小优化（SMO, Sequential Minimal Optimisation），它的附加优势不仅在于它是最具有竞争力的方法之一，还在于它易于实现。本章不可能做优化问题的穷尽讨论，在第7.8节提供了相关材料和在线软件的链接。

7.1 通用主题

与分类支持向量机相关的优化问题可以写为：

$$\begin{array}{ll} \text{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ & 0 \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, \ell \end{array} \tag{7.1}$$

这里$C = \infty$给出硬间隔情况或者二阶范数软间隔优化下核函数做过修正的情况，而$C < \infty$则给出了一阶范数软间隔的情况。对于回归问题，线性ε不敏感损失的问题则是：

$$\begin{array}{ll} \text{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} y_i \alpha_i - \varepsilon \sum_{i=1}^{\ell} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} & \sum_{i=1}^{\ell} \alpha_i = 0, -C \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, \ell \end{array} \tag{7.2}$$

这里令$C = \infty$并在核矩阵的对角位置增加一个常数，就是二次ε不敏感损失优化

问题。

函数 $W(\alpha)$和可行区域的凸性确保可以有效找到解。解满足 Karush-Kuhn-Tucker 互补条件。对分类最大间隔情况，它们是：

$$\alpha_i \left[y_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b \right) - 1 \right] = 0 \quad i = 1, \ldots, \ell$$

对二阶范数软间隔优化，它们是：

$$\alpha_i \left[y_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b \right) - 1 + \xi_i \right] = 0 \quad i = 1, \ldots, \ell$$

而对一阶范数软间隔优化：

$$\begin{aligned} &\alpha_i \left[y_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b \right) - 1 + \xi_i \right] = 0 && i = 1, \ldots, \ell \\ &(\alpha_i - C) \xi_i = 0 && i = 1, \ldots, \ell \end{aligned}$$

回归情况下，对二次ε不敏感损失函数，它们是：

$$\begin{aligned} &\alpha_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b - y_i - \varepsilon - \xi_i \right) = 0 && i = 1, \ldots, \ell \\ &\hat{\alpha}_i \left(y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b - \varepsilon - \hat{\xi}_i \right) = 0 && i = 1, \ldots, \ell \\ &\xi_i \hat{\xi}_i = 0, \alpha_i \hat{\alpha}_i = 0 && i = 1, \ldots, \ell \end{aligned}$$

而对线性ε不敏感损失函数，它们是：

$$\begin{aligned} &\alpha_i \left((\mathbf{w} \cdot \mathbf{x}_i) + b - y_i - \varepsilon - \xi_i \right) = 0 && i = 1, \ldots, \ell \\ &\hat{\alpha}_i \left(y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b - \varepsilon - \hat{\xi}_i \right) = 0 && i = 1, \ldots, \ell \\ &\xi_i \hat{\xi}_i = 0, \alpha_i \hat{\alpha}_i = 0 && i = 1, \ldots, \ell \\ &(\alpha_i - C) \xi_i = 0, (\hat{\alpha}_i - C) \hat{\xi}_i = 0 && i = 1, \ldots, \ell \end{aligned}$$

就像下面要讨论的，实践中，上述条件只能在一定的容忍程度内逼近。多数数值策略沿着这个思路，从任意一个可行点开始，通过迭代增加对偶目标函数的值，不离开可行区域，直到满足一个停止条件。另外一些思路是启发式的，每次仅作用于α_i的一个小的子集上来提高计算效率。有些类似的技术利用了问题的稀疏性，使得计算很大规模的数据集（成千上万个样例）成为可能。

从不同角度利用凸优化问题的性质可以得到不同的停止条件。一个选择是可行间隙在解中消失，通过检查这个量可以监视收敛与否。另一个选择是当对偶目标函数的增长小于某个预定的阈值时，计算停止。还有一个选择就是显式计算和监视 Karush-Kuhn-Tucker 条件来判断是否找到解。这三种停止条件的详细描述是：

1. 监视对偶目标函数的增长。对特定 SVM 优化问题，二次对偶目标函数在解上达到最大。监视函数的值，尤其是每一步中值的增长，这提供了最简单的停止条件。当目标函数 $W(\alpha)$的增长率低于某个给定的容忍值（比如10^{-9}）

时训练停止。遗憾的是，已经显示这个条件是不可靠的，在某些情况下结果很差。

2. 监视原问题的 Karush-Kuhn-Tucker 条件。它们是收敛的充分必要条件，因此提供了一个自然的条件。比如在分类情况下，对一阶范数软间隔优化一定要检查下面的条件：

$$0 \leqslant \alpha_i \leqslant C$$

$$y_i f(\mathbf{x}_i) \begin{cases} \geqslant 1 & \text{当} \alpha_i = 0 \\ = 1 & \text{当} 0 < \alpha_i < C \\ \leqslant 1 & \text{当} \alpha_i = C \end{cases}$$

注意这种情况下，不需要计算松弛变量 ξ_i。对二阶范数软间隔优化，条件是：

$$\alpha_i \geqslant 0$$

$$y_i f(\mathbf{x}_i) \begin{cases} \geqslant 1 & \text{当} \alpha_i = 0 \\ = 1 - \alpha_i / C & \text{当} \alpha_i > 0 \end{cases}$$

这种情况下，松弛变量由 $\xi_i = \alpha_i / C$ 隐式定义。自然这些条件在某个选定的容忍值下满足，这种情况下容忍值的一个好的选择是 10^{-2} 以内。

3. 刻画解的另一种方式是度量原始目标函数值和对偶目标函数值的间隙，它只有在优化点才消失。这个差别称为可行间隙，区别于优化问题中的对偶间隙，对偶间隙是原始解和对偶解的值的差别。对凸二次优化问题这个差别为零。在一阶范数软间隔优化的情况下，可行间隙可以像第 6.1.2 节描述的那样计算。首先令：

$$\xi_i = \max\left(0, 1 - y_i\left(\sum_{j=1}^{\ell} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b\right)\right)$$

这里 $\boldsymbol{\alpha}$ 是对偶问题的当前估计，选择 b 使得在某个 i 下 $C > \alpha_i > 0$ 有 $y_i f(\mathbf{x}_i) = 1$。原始目标和对偶目标的间隙可以给出如下：

$$\sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\sum_{j=1}^{\ell} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) \right) - 1 \right] + C \sum_{i=1}^{\ell} \xi_i =$$

$$= \sum_{i=1}^{\ell} \alpha_i - 2W(\boldsymbol{\alpha}) + C \sum_{i=1}^{\ell} \xi_i$$

这里 $W(\boldsymbol{\alpha})$ 是对偶目标。比率：

$$\frac{\text{原始目标-对偶目标}}{\text{原始目标+1}} = \frac{\sum_{i=1}^{\ell}\alpha_i - 2W(\boldsymbol{\alpha}) + C\sum_{i=1}^{\ell}\xi_i}{W(\boldsymbol{\alpha}) + \sum_{i=1}^{\ell}\alpha_i - 2W(\boldsymbol{\alpha}) + C\sum_{i=1}^{\ell}\xi_i + 1}$$
$$= \frac{\sum_{i=1}^{\ell}\alpha_i - 2W(\boldsymbol{\alpha}) + C\sum_{i=1}^{\ell}\xi_i}{\sum_{i=1}^{\ell}\alpha_i - W(\boldsymbol{\alpha}) + C\sum_{i=1}^{\ell}\xi_i + 1} \tag{7.3}$$

为进程提供了一个有用度量，检查它是否小于一个值比如10^{-3}，可以用做停止条件。

对最大间隔和二阶范数间隔松弛优化（通过在核矩阵的对角上增加一个常数来实现），可以在每次迭代 t 中引入等于最大 α_i 的盒约束$C_t = \max_i\left(\alpha_i^t\right) + 1$。这意味着算法的运行不会受影响，每一次迭代 $\boldsymbol{\alpha}$ 的当前值可以视做对应着盒约束优化的一个可行解。对这个问题，可使用上面的方程计算可行间隙，其中 C 要足够大，$C > \max_i\left(\alpha_i^*\right)$，这里 $\boldsymbol{\alpha}^*$ 是无约束问题的最优解，两个解是一致的。

评注 7.1 注意停止条件的容忍等级很重要。达到高的准确程度是很耗时的，而且在预测精度上不会有重要的优势。因此实践中需要设置适当的容忍等级来确保逼近优化条件。

评注 7.2 停止条件的一个重要结果是促进使用启发算法来提高收敛速度：比如通过作用到对可行间隙更有贡献的点来更快地找到解。其中，这些点可以是下式中数值较大的点：

$$\alpha_i\left[y_i\left(\sum_{j=1}^{\ell}y_j\alpha_j K(\mathbf{x}_j,\mathbf{x}_i)\right) - 1\right] + C\xi_i$$

或者进行某些操作使得对偶目标函数有更大的增长。下几节设计更有效的训练算法时会利用这些思路。

评注 7.3 应该注意如果核矩阵仅仅是半正定的，那么即使解 $\mathbf{w}$ 是惟一的，它以 $\boldsymbol{\alpha}$ 为项的扩展也许不惟一。

本章的剩余各节，将首先讨论一个没有优化偏置的简单梯度上升算法。这提供了一个台阶来进一步介绍一系列重要概念，这些概念在后面各节更复杂的算法中将要用到。

7.2 简单解：梯度上升算法

凸优化问题最简单的数值解方法是梯度上升法，有时称为最速上升法。这个算

法从解的初始估计α^0开始，沿着最速上升的路径迭代更新，在$t+1$次，沿着$W(\alpha)$在位置α^t的梯度方向移动。每次迭代更新的方向由最速上升策略决定但步长仍然是固定的。更新的步长称为学习率。

在序列或随机版本中，每次仅为一个样例计算梯度来近似实现上面的策略，因此通过增量：

$$\delta\alpha_i^t = \eta\frac{\partial W(\alpha^t)}{\partial\alpha_i}$$

更新单个成分α_i^t，这里参数η是学习率。如果恰当选择η，目标函数将单调上升，平均方向逼近局部梯度。可以调整η使其成为一个时间函数，或者是正在学习的输入模式的函数，目的是为了改进收敛。η的选择是个双刃剑，η过大会使得系统振荡不能收敛到解，过小则使得收敛很慢。沿着这个思路意味着每次迭代的方向平行于某个ℓ轴，也就是所知的先验。算法决定步长以及更重要的步长的符号。进一步，一个可行的自由度是所更新点的次序的选择，因为有些点在通向解的过程中沿着路径会引起更大的移动。这将在下面进一步讨论。

看待相同算法的另一种方式是固定其他所有变量，只通过一个变量使$W(\alpha)$迭代增加。因此一个多维问题可以简化为一个一维问题的序列。全局最大值的惟一性保证适当选择η，算法总可以找到解。从速度的角度看这样的策略不是最优的，但对几千个点的数据集十分适合。还有一个优势，就是实现非常方便。后面将看到，另一个主要的优势是每次仅作用于一个训练点，因此不需要同时在内存中存储所有的数据。

使用这个方法，问题部分源于线性约束：

$$\sum_{i=1}^{\ell}\alpha_i y_i = 0 \tag{7.4}$$

这是在决策函数中优化偏置时得出的。这个约束定义了ℓ维参数空间中的一个超平面，并且将可行区域限制到一个交叉区域。这个交叉区域由两部分形成，一部分是这个超平面；另一部分在一阶范数软间隔优化中是参数 C 的超管道，而在最大间隔和二阶范数软间隔优化中是位于正的象限。实现这个约束的一个自然的策略是确保当前解不离开可行区域。遗憾的是，每次只更新一个成分是不可能保证的：如果第t次更新后，方程（7.4）的约束是满足的，但是在一个α_i上做一次非平凡的更新后，它就不再满足了。为了保证解不离开可行区域，要同时更新的最小乘子数目是2，由此形成了第7.5节SMO算法的基础。其他策略也存在，比如实现约束：

$$-\zeta_t \leqslant \sum_{i=1}^{\ell}\alpha_i^t y_i \leqslant \zeta_t$$

这里每次迭代ζ_t是递减的。本节将讨论偏置 b 是一个固定值的情况下的算法，因此不需要实现等式约束。

评注 7.4 预先固定偏置看起来是有局限性的。如果考虑输入空间 X 嵌入到一个额外增加一维的空间 $\hat{X}$，其中对某些固定的τ值新的向量可以表示为 $\hat{\mathbf{x}} = (\mathbf{x}, \tau)$，那么在通过单位权重向量 $\mathbf{w}$ 和偏置 b 表示的空间 X 上的线性函数，等价于空间$\hat{X}$上的$\hat{\mathbf{w}} = (\mathbf{w}, b/\tau)$和零偏置表示的函数，也就是：

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = \langle \hat{\mathbf{w}} \cdot \hat{\mathbf{x}} \rangle$$

注意对一个训练集：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

的非平凡分类，一定有$b \leqslant R = \max_{1 \leqslant i \leqslant \ell} (\|\mathbf{x}_i\|)$。增加一维到核 K 形成的特征空间等价于在核上增加τ^2来修正核：

$$\hat{K}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z}) + \tau^2$$

这个思路仅有的缺点是扩充空间中分开超平面的几何间距通常小于原始空间。如果τ选择得不恰当，这个差别将非常大，会影响算法的收敛和所得分类器的泛化性能。假定$\mathbf{w}$是归一化的，数据集 $\hat{S}$ 上新的权重向量 $\hat{\mathbf{w}}$ 的函数间隔等价于 S 上 $\mathbf{w}$ 的几何间隔 γ。因此 $\hat{\mathbf{w}}$ 的几何间隔是：

$$\begin{aligned} \gamma \|\hat{\mathbf{w}}\|^{-1} &= \gamma \left(1 + b^2/\tau^2\right)^{-1/2} \\ &\leqslant \gamma \left(1 + R^2/\tau^2\right)^{-1/2} \end{aligned}$$

空间 X（见第 4 章定理 4.16）中度量宽打散维的量是比率：

$$\frac{\max_{1 \leqslant i \leqslant \ell} \left(\|\mathbf{x}_i\|^2\right)}{\gamma^2} = \frac{R^2}{\gamma^2}$$

在$\hat{X}$中这个比率变为：

$$\frac{\max_{1 \leqslant i \leqslant \ell} \left(\|\mathbf{x}_i\|^2\right) + \tau^2}{\gamma^2 \left(1 + R^2/\tau^2\right)^{-1}} = \frac{\left(R^2 + \tau^2\right)\left(1 + R^2/\tau^2\right)}{\gamma^2}$$

取$\tau = R$时上式右侧得到最小值$4R^2/\gamma^2$。因此τ的一种安全选择是等于 R，这仅使得在宽打散维上的界增加一个系数 4。

如果令偏置为固定值，优化问题的对偶表示变为：

$$\begin{array}{ll}\text{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} & 0 \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, \ell\end{array}$$

$W(\boldsymbol{\alpha})$梯度的第 i 个成分是：

$$\frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i} = 1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

因此可以一个固定的学习率 η 使用简单的迭代更新规则：

$$\alpha_i \leftarrow \alpha_i + \eta \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i}$$

最大化 $W(\boldsymbol{\alpha})$，并且在一个正象限内同时更新所有 α_i 的值。当 α_i 为负值时，可以令其为零，也就是运行下面的更新规则：

$$\alpha_i \leftarrow \max\left(0, \alpha_i + \eta \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i}\right)$$

类似地，当 α_i 的上界由软间隔技术控制时，一旦它超过 C，可以令其为 C，也就是运行更新：

$$\alpha_i \leftarrow \min\left(C, \max\left(0, \alpha_i + \eta \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i}\right)\right)$$

这就是所谓投影方法。

在无偏置情况下训练 SVM 的简单方法如表 7.1 所示。注意每一个训练样例有独立的学习率 η_i。

表 7.1 具有一阶软间隔的简单在线算法

给定训练集 S 和学习率 $\boldsymbol{\eta} \in (\mathbb{R}^+)^{\ell}$
$\boldsymbol{\alpha} \leftarrow \mathbf{0}$
重复
  for $i = 1$ to ℓ
    $\alpha_i \leftarrow \alpha_i + \eta_i\left(1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)\right)$
    if $\alpha_i < 0$ then $\alpha_i \leftarrow 0$
    else
    if $\alpha_i > C$ then $\alpha_i \leftarrow C$
  end for
直到满足停止条件
返回 $\boldsymbol{\alpha}$

在两个方面这个算法不能实现严格的梯度上升。首先，对每个样本使用不同的学习率，使得梯度的方向偏离。第二，如果希望严格地实现梯度，应该在 for 循环中建立一个新的向量α^{new}，然后在每次循环结束时设$\alpha = \alpha^{new}$。实际上，获得α_i的值后，它的使用是方便有效的。这就是随机梯度上升算法，与解决矩阵方程的连续过度松弛技术有联系。作为停止条件，可以使用上面给出的任何一个，也就是监视 Karush-Kuhn-Tucker 条件、可行间隙或者目标函数$W(\alpha)$简单的增长比率。事实上，算法的平稳条件对应着问题的 Karush-Kuhn-Tucker 条件。适当选择η_i，算法会收敛。更精确地说，优化α_i的二次函数，使其导数为 0 的更新规则是：

$$\hat{\alpha}_i \leftarrow \alpha_i + \frac{1}{K(\mathbf{x}_i, \mathbf{x}_i)}\left(1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)\right)$$

因为：

$$\begin{aligned}\frac{\partial W(\hat{\alpha})}{\partial \alpha_i} &= 1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\quad - y_i y_i K(\mathbf{x}_i, \mathbf{x}_i) \frac{1}{K(\mathbf{x}_i, \mathbf{x}_i)}\left(1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)\right) \\ &= 0\end{aligned}$$

因此，假定$0 \leqslant \hat{\alpha}_i \leqslant C$，选择：

$$\eta_i = \frac{1}{K(\mathbf{x}_i, \mathbf{x}_i)}$$

可以实现最大增益，同时收敛的充分条件是$0 < \eta_i K(\mathbf{x}_i, \mathbf{x}_i) < 2$。当这个更新由可行区域的边界约束时，步长会越来越短，面增益会相应越来越小，但保持正值。尽管这明显是不可靠的，但是使用上面的η_i的值，存在常数$\mu, \delta \in (0,1)$和$\tau \in (0,1)$满足：

$$\left\|\alpha^t - \alpha^*\right\| \leqslant \mu \delta^t$$

和

$$W(\alpha^*) - W(\alpha^{t+1}) \leqslant \tau\left(W(\alpha^*) - W(\alpha^t)\right)$$

尽管实践中算法的性能变化很大，但这样的收敛比率是确定的。改进收敛比率的一种重要方式是增加自由度，使得参数更新次序变化。表 7.1 显示参数是序列更新的，但很明显这个次序可以在新的循环开始的时候变化；或者事实上如果没有点被忽略，更新可以在任意点上迭代进行。很明显如果辨识出那些对对偶目标函数的增长贡献大的点，则可以优先选择它们。这提示可以利用启发式方法选择那些违反 Karush-

Kuhn-Tucker 条件的点。算法的外循环浏览训练集寻找违反 Karush- Kuhn-Tucker 条件的点，选择任一个点进行更新。为了增加找到 Karush-Kuhn-Tucker 条件违反点的机会，外循环考虑那些相应参数满足 $0 < \alpha_i < C$ 的点，因为这意味着它们的值不在可行性区域的界上，仅当所有这些点在容忍等级内满足 Karush-Kuhn-Tucker 条件，一个在所有训练集上完整的循环才能重新开始。对这个思路的简单改进将在评注 7.8 中介绍。

评注 7.5　每次优化一个对偶变量的算法就是优化文献中所述的 Hildreth 方法，而在机器学习中通常指的是核 Adatron 算法。如果忽略核的使用，上述算法等价于训练单个神经元的 Adatron 算法。

评注 7.6　在线梯度算法最近的变体使用了附加参数 $\omega \in (0,2)$ 来实现连续的过度松弛，剪辑前的更新是：

$$\alpha_i \leftarrow \alpha_i + \frac{\omega}{K(\mathbf{x}_i,\mathbf{x}_i)}\left(1 - y_i\sum_{j=1}^{\ell}\alpha_j y_j K(\mathbf{x}_i,\mathbf{x}_j)\right)$$

在这个方法的实验中，选择$\omega = 1$，它等价于表 7.1 的算法中选择 $\eta_i = (K(\mathbf{x}_i,\mathbf{x}_i))^{-1}$。

评注 7.7　可以在初始化过程中将核矩阵 $K(\mathbf{x}_i,\mathbf{x}_j)$ 的值放入内存来节省训练时间，因为这些值需要重复使用，而且核函数的计算很费时。这种方法在某种程度上牺牲了算法内在的在线特性，实践中，训练集可以是中等大小。下面各节中的多数标准算法采用了这个算法，有时它还与第 7.4 节的子集选择方法结合。对较大的数据集，可以在每次迭代中重新计算核函数，这降低了存储空间的复杂性，但增加了训练时间。对中等样本数目（见第 8 章的例子）一般采用随机梯度上升算法，结合样本启发式选择算法，它对大规模的数据集也是有效的，而其他算法会失败。在下一个评注中将描述启发式算法，它成功地处理了一百万个点的数据集。

评注 7.8　启发式选择要处理的数据点的次序对收敛率有很重要的影响。一个特别简单的排序是使用参数 α_i 的大小。这与上面描述的在支持向量当前估计上的二段式策略结合，仅当这种条件下最优实现时，才再次考虑那些 $\alpha_i = 0$的点。这提示了可以仅在内存保存那些非零的 α_i。这些是支持向量的当前估计。它们按照 α_i 的大小的升序循环处理，直到目标不再有进一步的变化。此时可以在数据集的剩余点上进行迭代，这产生了新的支持向量，并开始新的循环。第 7.8 节给出将这个方法成功用于一百万个点的数据集的支持向量优化的论文。

评注 7.9　注意第 2 章描述的感知机算法可视为一种梯度上升算法。代价函数是 $\sum_{i=1}^{\ell}\xi_i$，这里 ξ_i 定义为$\max\left(0, -y_i((\mathbf{w}\cdot\mathbf{x}_i)+b)\right)$。这个函数有时称为铰链损失，等价

于$\varepsilon = 0$的线性ε不敏感损失。近几年关于梯度上升算法的较多理论研究信息可以参阅第7.8节。

评注 7.10 上面描述的算法解决了一阶范数软间隔优化问题。很明显，如果忽略$\alpha_i \leqslant C$的约束，这个算法覆盖了最大间隔算法。如果希望解决二阶软间隔优化，可以在核矩阵上简单地增加一个对角平移$\mathbf{K} \leftarrow \mathbf{K} + \frac{1}{C}\mathbf{I}$。

当然，本节列出的简单算法会受制于许多与简单梯度上升相关的问题：在有的数据集上很慢，在收敛前振荡，等等。然而，它们的概念和计算简单性使其在小数据集的应用上成为支持向量机第一个实现的理想候选算法。更高级的技术将在下面的各节中讨论。

7.3 通用技术和软件包

到目前为止，已经开发了若干种优化技术，其中许多可以直接应用到二次规划中。牛顿方法、共轭梯度、原对偶内点方法不仅可以直接应用到支持向量机的情况中，而且当目标函数的特定结构给定时它们还可以简化。从概念上讲，它们与上面描述的简单的梯度上升方法没有很大差别，所有这些方法都是通过迭代找到最大值，但是这里介绍的三种方法更加有效，从本质上讲是因为每个步骤的方向和长度都以一种更复杂的方式选择。因为问题的规模，要将更多的注意力放到计算方面。多数方法需要将核矩阵存储到内存，意味着空间复杂度表示为样本数目的二次方。对大规模问题，这些方法都是低效的，因而需要与第7.4节的分解技术结合使用。本章不可能描述过多的方法，综述的链接将在第7.8节给出。

这些技术的一个主要优势是容易理解，并且广泛地存在于大量商业软件和自由软件包中，其中一些可以通过因特网获得。在特别定制的算法出现前，就是这些软件包用于支持向量机的实现。一个最普遍的选择是MINOS软件包，它来自Stanford大学的优化实验室，使用了一种混合策略；另一个标准的选择是LOQO，它使用了原对偶内点方法。相反的是MATLAB优化工具箱提供的二次规划子模块很通用，但是模块quadprog比qp要好很多。这些软件包以及其他的链接将在第7.8节给出。

最后，一个非常便利的方案是使用已经存在的支持向量机软件包，如Joachims的$\text{SVM}^{\text{light}}$，London大学Royal Holloway分校的软件包和其他可以免费从因特网上获取的软件包。类似地，高斯过程的软件包也可以免费从因特网上获取。文献和在线软件的链接将在第7.8节给出。

7.4 块与分解

前一节描述的技术没有随机梯度上升法的在线优势，因为这些方法中需要数据以核矩阵的形式存储在内存中。训练问题的复杂度随着矩阵的规模上升，使得能处理的数据集的规模在几千点以内。

对较大规模的问题，要在优化中利用所谓“活动集”或“工作集”方法带来的优势，这种方法可描述为：如果事先知道哪个约束是积极的，就有可能放弃所有非积极约束，从而简化问题。这导出几个策略，都是建立在如何猜测活动集上的，并将训练限制在这个猜测集上。迭代启发逐渐建立活动集是最常用的。尽管这些技术通常是启发式的，但是每步都减少可行间隙或增加对偶目标函数的事实可以确保算法最终的收敛性。

最简单的启发称为块。从数据的任意子集或块出发，在该部分数据上使用通用的优化算法训练数据。算法保留了支持向量而丢弃了其他点，然后使用找到的假设去检测数据剩余的点。违反 Karush-Kuhn-Tucker 条件最严重的 M（系统的参数）个点加入到前面问题的支持向量中去形成一个新的块。算法是迭代的，为每个带有上个阶段输出值的子问题初始化 α，最后当某个停止条件满足时，迭代结束。在特定阶段优化的数据块有时称为工作集。工作集的规模尽管有时会减小，但通常是增加的，直到学习器在代表积极约束的支持向量上训练到最后一个迭代。算法的伪码在表 7.2 中给出。

表 7.2 通用工作集方法的伪码

```
给定训练集 S
α ← 0
选择任意工作集 Ŝ ⊂ S
重复
    在 Ŝ 上求解优化问题
    从不满足 Karush-Kuhn-Tucker 条件的数据中选择新的工作集
直到满足停止条件
返回 α
```

启发式算法假定支持向量集的核矩阵适合内存并且可以输入到所使用的优化软件包。一般地，如果问题不是稀疏的或者问题规模很大，支持向量的活动集仍然会很大，以至于不能在优化模块中处理。可以使用更高级的算法，比如分解，它受到工作集方法中块方法的启发。分解算法只更新乘子 α_i 的一个固定大小的子集，其他保持不变。因此，每当一个新点加入到工作集，另一个点要被移除。在这个算法中，

目标不是辨识所有的积极约束并在它们上面运行优化算法，而是每次在数据的一个小的子集上优化全局问题。在这个系统中，同前面所述，算法内核是由一些通用的二次规划优化算法提供，可能就是许多可行软件包中的一个。

尽管还没有给出这些方法收敛性的理论证明，但在实践中它们工作得很好，使得处理成千上万个点的数据集成为可能。

选择数据集方式的重点在于，使得对应的二次规划子问题的优化成为整个目标函数的改进。有几个启发式方法可用于此，每次迭代中选择工作集的一种有效的启发式方法是使用停止条件：比如包含那些对可行间隙贡献最大的点，或者等价地说是包含那些违反 Karush-Kuhn-Tucker 条件最严重的点。

7.5 序贯最小优化算法

序贯最小优化（SMO）算法是将分解算法思想推向极致得出的，而每次迭代仅优化两个点的最小子集。这项技术的威力在于两个数据点的优化问题可以获得解析解，从而不需要将二次规划优化算法作为算法一部分。

对条件$\sum_{i=1}^{\ell}\alpha_i y_i = 0$的需要在迭代中实现，意味着每步能优化的乘子最小个数为2：无论何时一个乘子被更新，至少需要调整另一个乘子来保持条件成立。

每步 SMO 选择两个元素α_i和α_j共同优化，在其他参数固定的前提下，找到这两个元素参数的最优值，并更新相应的α向量。这两个点的选择是启发式的，而这两个点的乘子的优化可以获得解析解。尽管需要更多的迭代才收敛，但每个迭代需要很少的操作，因此算法在整体上的速度有数量级的提高。包括收敛时间在内，算法其他的特征是没有矩阵操作，它不需要在内存存储核矩阵；它不需要其他优化软件包；并且它很容易实现。注意因为标准 SMO 不需要存储核矩阵，核矩阵的引入可能会获得更进一步的速度提高，代价是增加了空间复杂度。

7.5.1 两点解析解

不损失一般性，假定选定的两个元素是α_1和α_2。要为这两个参数计算新的值，可以看出，为了不违反线性约束$\sum_{i=1}^{\ell}\alpha_i y_i = 0$，乘子的新值必须在一条直线上：

$$\alpha_1 y_1 + \alpha_2 y_2 = \text{常数} = \alpha_1^{old} y_1 + \alpha_2^{old} y_2$$

这条线是在(α_1, α_2)的空间，并在$0 \leqslant \alpha_1, \alpha_2 \leqslant C$的盒子约束中。约束目标函数到一条直线上所得到的一维问题有解析解。

不损失一般性，用该算法首先可以计算α_2^{new}，并使用它得到α_1^{new}。盒子约束$0 \leqslant \alpha_1, \alpha_2 \leqslant C$和线性等式约束，在$\alpha_2^{new}$的可行值上提供了一个更严格的约束：

$$U \leqslant \alpha_2^{new} \leqslant V$$

这里如果$y_1 \neq y_2$，则：

$$\begin{aligned} U &= \max\left(0, \alpha_2^{old} - \alpha_1^{old}\right) \\ V &= \min\left(C, C - \alpha_1^{old} + \alpha_2^{old}\right) \end{aligned} \tag{7.5}$$

而如果$y_1 = y_2$，则：

$$\begin{aligned} U &= \max\left(0, \alpha_1^{old} + \alpha_2^{old} - C\right) \\ V &= \min\left(C, \alpha_1^{old} + \alpha_2^{old}\right) \end{aligned} \tag{7.6}$$

评注 7.11 下面的定理将使用上面给出的U和V的定义。同样要引入几个符号简化定理和证明的表述。使用$f(\mathbf{x})$来表示在学习的特定阶段值α和b所决定的当前假设。令：

$$E_i = f(\mathbf{x}_i) - y_i = \left(\sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) + b\right) - y_i \quad i = 1, 2 \tag{7.7}$$

是训练点$\mathbf{x}_1$或$\mathbf{x}_2$上的函数输出和目标分类的差别。注意，即使点正确分类，这个值可能也很大。比如，若$y_1 = 1$，函数输出是$f(\mathbf{x}_1) = 5$，分类是正确的，但是$E_1 = 4$。需要的一个进一步的量是目标函数在对角线上的二次导数，可以表示为$-\kappa$，这里：

$$\kappa = K(\mathbf{x}_1, \mathbf{x}_1) + K(\mathbf{x}_2, \mathbf{x}_2) - 2K(\mathbf{x}_1, \mathbf{x}_2) = \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|^2 \tag{7.8}$$

其中$\phi(\cdot)$是原始空间到特征空间的映射。

现在给出并证明下面的定理。

定理 7.12 当α_1和α_2允许改变时，优化问题（7.1）的目标函数的最大值，可以通过计算下面的量得到：

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\kappa}$$

剪辑它来实现约束$U \leqslant \alpha_2^{new} \leqslant V$：

$$\alpha_2^{new} = \begin{cases} V & \text{当} \alpha_2^{new,unc} > V \\ \alpha_2^{new,unc} & \text{当} U \leqslant \alpha_2^{new,unc} \leqslant V \\ U & \text{当} \alpha_2^{new,unc} < U \end{cases}$$

这里E_i由方程（7.7）给出，κ由方程（7.8）给出，U和V由方程（7.5）或方程（7.6）给出。可以从α_2^{new}中得到α_1^{new}的值：

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2(\alpha_2^{old} - \alpha_2^{new})$$

证明 定义：

$$v_i = \sum_{j=3}^{\ell} y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i) - \sum_{j=1}^{2} y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - b \quad i = 1,2$$

考虑将 α_1 和 α_2 的函数作为目标：

$$\begin{aligned} W(\alpha_1, \alpha_2) &= \alpha_1 + \alpha_2 - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ &\quad -y_1 y_2 K_{12}\alpha_1\alpha_2 - y_1\alpha_1 v_1 - y_2\alpha_2 v_2 + \text{常数} \end{aligned}$$

这里 $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, $i,j = 1,2$。注意约束 $\sum_{i=1}^{\ell} \alpha_i^{old} y_i = \sum_{i=1}^{\ell} \alpha_i y_i = 0$ 意味着条件：

$$\alpha_1 + s\alpha_2 = \text{常数} = \alpha_1^{old} + s\alpha_2^{old} = \gamma$$

这里 $s = y_1 y_2$。利用这个方程可以用 α_2^{new} 计算 α_1^{new}，此约束下目标函数可写为：

$$\begin{aligned} W(\alpha_2) &= \gamma - s\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}(\gamma - s\alpha_2)^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ &\quad -sK_{12}(\gamma - s\alpha_2)\alpha_2 - y_1(\gamma - s\alpha_2)v_1 - y_2\alpha_2 v_2 + \text{常数} \end{aligned}$$

驻点满足：

$$\begin{aligned} \frac{\partial W(\alpha_2)}{\partial \alpha_2} &= 1 - s + sK_{11}(\gamma - s\alpha_2) - K_{22}\alpha_2 \\ &\quad + K_{12}\alpha_2 - sK_{12}(\gamma - s\alpha_2) + y_2 v_1 - y_2 v_2 \\ &= 0 \end{aligned}$$

由此得出：

$$\begin{aligned} \alpha_2^{new,unc}(K_{11} + K_{22} - 2K_{12}) &= 1 - s + \gamma s(K_{11} - K_{12}) + y_2(v_1 - v_2) \\ &= y_2(y_2 - y_1 + \gamma y_1(K_{11} - K_{12}) + v_1 - v_2) \end{aligned}$$

因此：

$$\begin{aligned} \alpha_2^{new,unc}\kappa y_2 &= y_2 - y_1 + f(\mathbf{x}_1) - \sum_{j=1}^{2} y_j\alpha_j K_{1j} + \gamma y_1 K_{11} \\ &\quad -f(\mathbf{x}_2) + \sum_{j=1}^{2} y_j\alpha_j K_{2j} - \gamma y_1 K_{12} \\ &= y_2 - y_1 + f(\mathbf{x}_1) - f(\mathbf{x}_2) \\ &\quad +y_2\alpha_2 K_{11} - y_2\alpha_2 K_{12} + y_2\alpha_2 K_{22} - y_2\alpha_2 K_{12} \\ &= y_2\alpha_2\kappa + (f(\mathbf{x}_1) - y_1) - (f(\mathbf{x}_2) - y_2) \end{aligned}$$

给出：

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\kappa}$$

最后，必要的情况下剪辑 $\alpha_2^{new,unc}$，确保它在区间$[U, V]$内。

7.5.2 启发式选择算法

为了提高收敛速度，可以根据点在求解过程中的贡献选择其中两个作为子集优化目标函数。如果实现选择策略所需要的计算量小于节省下来的迭代所需的计算量，就在收敛过程中获得了收益。

选择的停止条件能指示出哪些点更易于对收敛做出更大贡献。比如，如果监视可行间隙，一个自然的选择是优化那些最违反 Karush-Kuhn-Tucker 条件的点，它们对间距贡献最大（见第 7.1 节的停止条件 3）。在每次迭代中计算每个点的 Karush-Kuhn-Tucker 条件花费高昂，廉价的启发式选择会带来较好的整体性能。

SMO 使用两个条件来选择两个活动点，确保目标函数在优化过程中有较大的增长。分别有两个单独的启发式算法用来选择第一个点和第二个点。

第一个选择的启发式算法 第一个点$\mathbf{x}_1$从最违反 Karush-Kuhn-Tucker 条件的那些点中选取。算法的外循环浏览数据集寻找违反 Karush-Kuhn-Tucker 条件的点，并使用任意一个点来更新。当找到一个这样的点，用第二个选择的启发式方法选择第二个点，更新各自乘子的值。然后外循环重新寻找新的 Karush-Kuhn-Tucker 违反者。为了提高找到 Karush-Kuhn-Tucker 违反点的机会，外循环浏览那些对应参数满足 $0 < \alpha_i < C$ 的点，这意味着它的值不是在可行区域的边界上，只有当这些点在特定的容忍等级满足 Karush-Kuhn-Tucker 条件，在整个数据集上的完整循环才算结束，然后进入下一个数据集。

第二个选择的启发式算法 第二个点 $\mathbf{x}_2$ 的选择一定要按这样的方式，在 α_1, α_2 上的更新引起大的变化，使得对偶目标有大的增长。为了不进行过多的计算就找到一个好点，一种快速的启发式方法是选择最大化量 $|E_1 - E_2|$ 的点为 $\mathbf{x}_2$，这里 E_i 在定理 7.12 中定义。如果 E_1 是正的，SMO 选择一个具有最小误差 E_2 的样例 $\mathbf{x}_2$；而如果 E_1 是负的，SMO 选择一个具有最大误差 E_2 的样例。内存保留数据集每一个不在边界上的点的误差，可以减少计算量。如果这个选择没有在对偶目标上产生大的增长，SMO 会轮流尝试每个非边界点。遍历非边界点和整个数据集的循环从各自列表的随机位置开始，这样就不会在两者中任何一个开始的时候就引入偏置。

John Platt 的 SMO 分类算法的伪码在附录 A 给出。注意在第 7.8 节给出了实现

SMO 的在线软件和 SMO 的完整详细描述的文献的链接。

评注 7.13 注意 SMO 算法明显使用了参数 C，初看起来它只能应用到一阶范数软间隔优化问题。然而可以将 C 视为无穷大，运行 SMO，从而简化在区间 $[U,V]$上的约束，它仅提供了一个α_2^{new}的下界，当 $y_1 \neq y_2$时：

$$U = \max\left(0, \alpha_2^{old} - \alpha_1^{old}\right)$$

当 $y_1 = y_2$时：

$$\begin{aligned} U &= 0 \\ V &= \alpha_1^{old} + \alpha_2^{old} \end{aligned}$$

评注 7.14 SMO 算法没有提供方法来选择偏置，但却使用它来计算 $E_i, i = 1,2$。很明显，这并不影响算法，因为无论 b 的值是什么，两个 E_i受到相同的影响，因此所得更新和它们的差决定的更新是相同的。所以计算过程中 b 的值可以设为 0，算法收敛后可以使用本章开始就介绍的 Karush-Kuhn-Tucker 条件对特定优化问题设置 b 的值。注意，在评价停止条件时可能要计算 b 的值。

评注 7.15 第 7.1 节的停止条件 3 可用来评价收敛。注意这需要设置偏置的值。作为部分计算的标示，b 的选择应该考虑那些 α_i 的值满足$0 < \alpha_i < C$的点。原始的 SMO 算法在更新过的点上估计偏置。如果两个点不满足所需的不等式，会导致可行间隙的过估计。

评注 7.16 注意 SMO 不能直接应用到偏置固定的情况下，因为α_2^{new}的选择是使用偏置作为变量得到的约束做出的。对固定的偏置，SMO 算法简化为表 7.1 描述的算法，并且有：

$$\eta_i = (K(\mathbf{x}_i, \mathbf{x}_i))^{-1}$$

对于回归，SMO 算法可以从问题（7.2）给出的优化问题再次更新 α_1 和 α_2。方程编码四个独立的问题与两个参数的符号相关。这里在 $\boldsymbol{\alpha}$ 向量上的约束没有涉及分类问题，所以 α_2 的区间由下式给出：

$$\begin{aligned} U &= \max\left(C_U^2, \alpha_1^{old} + \alpha_2^{old} - C_V^1\right) \\ V &= \min\left(C_V^2, \alpha_1^{old} + \alpha_2^{old} - C_U^1\right) \end{aligned} \tag{7.9}$$

这里四个问题的参数 C_U^i, C_V^i 有不同的设置，在下表给出：

	$\alpha_i \geqslant 0$	$\alpha_i \leqslant 0$
C_U^i	0	$-C$
C_V^i	C	0

(7.10)

评注 7.17 下面的定理将使用上面给出的 U 和 V 的定义。同样要引入几个符号简化定理和证明的表述。使用 $f(\mathbf{x})$ 表示在学习的特定阶段 α 和 b 所决定的当前假设。令：

$$E_i = f(\mathbf{x}_i) - y_i = \left(\sum_{j=1}^{\ell} \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b\right) - y_i \quad i = 1,2 \qquad (7.11)$$

是训练点 $\mathbf{x}_1$ 或 $\mathbf{x}_2$ 上的函数输出值和目标值的差别。

现在证明下面的定理，这个定理展示了如何将 SMO 应用到回归问题。注意这个定理有效地给出了四个更新规则，对应着当前值 α_1 和 α_2 空间的四个象限。重要的是要在优化后从包含当前值的一个象限中寻找新的值。如果当前值落在不止一个象限内，应该对每个象限做优化，选择对目标函数有着较大增长的象限。

定理 7.18 当 α_1 和 α_2 在包含两个值的特定象限中允许改变时，优化问题（7.1）的目标函数的最大值，可以通过计算下面这个量得到：

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{(E_1 - E_2) - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1))}{\kappa}$$

剪辑它来实现约束 $U \leqslant \alpha_2^{new} \leqslant V$：

$$\alpha_2^{new} = \begin{cases} V & \text{当 } \alpha_2^{new,unc} > V \\ \alpha_2^{new,unc} & \text{当 } U \leqslant \alpha_2^{new,unc} \leqslant V \\ U & \text{当 } \alpha_2^{new,unc} < U \end{cases}$$

这里符号函数的值由选择的象限决定。E_i 由方程（7.11）给出，κ 由方程（7.8）给出，而 U 和 V 由方程（7.9）或方程（7.10）给出。从 α_2^{new} 中得到 α_1^{new} 值：

$$\alpha_1^{new} = \alpha_1^{old} + \alpha_2^{old} - \alpha_2^{new}$$

证明 定义

$$v_i = \sum_{j=3}^{\ell} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i) - \sum_{j=1}^{2} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - b \quad i = 1,2$$

考虑用 α_1 和 α_2 的函数作为目标函数：

$$\begin{aligned} W(\alpha_1, \alpha_2) \quad = \quad & y_1\alpha_1 + y_2\alpha_2 - \varepsilon(|\alpha_1| + |\alpha_2|) - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ & - K_{12}\alpha_1\alpha_2 - \alpha_1 v_1 - \alpha_2 v_2 + \text{常数} \end{aligned}$$

这里 $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$，$i, j = 1,2$。注意约束 $\sum_{i=1}^{\ell} \alpha_i^{old} = \sum_{i=1}^{\ell} \alpha_i = 0$ 意味着条件：

$$\alpha_1 + \alpha_2 = \text{常数} = \alpha_1^{old} + \alpha_2^{old} = \gamma$$

利用这个方程可以用α_2^{new}计算α_1^{new}，此约束下的目标函数可写为：

$$\begin{aligned} W(\alpha_2) &= y_1\gamma - y_1\alpha_2 + y_2\alpha_2 - \varepsilon\left(|\alpha_1| + |\alpha_{2i}|\right) - \frac{1}{2}K_{11}(\gamma-\alpha_2)^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ &\quad -K_{12}(\gamma-\alpha_2)\alpha_2 - (\gamma-\alpha_2)v_1 - \alpha_2 v_2 + \text{常数} \end{aligned}$$

驻点满足：

$$\begin{aligned} \frac{\partial W(\alpha_2)}{\partial \alpha_2} &= y_2 - y_1 - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1)) + K_{11}(\gamma-\alpha_2) - K_{22}\alpha_2 \\ &\quad +K_{12}\alpha_2 - K_{12}(\gamma-\alpha_2) + v_1 - v_2 \\ &= 0 \end{aligned}$$

由此得出：

$$\begin{aligned} \alpha_2^{new,unc}(K_{11}+K_{22}-2K_{12}) &= y_2 - y_1 - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1)) \\ &\quad +\gamma(K_{11}-K_{12}) + v_1 - v_2 \\ &= y_2 - y_1 + \gamma(K_{11}-K_{12}) + v_1 - v_2 \end{aligned}$$

因此：

$$\begin{aligned} \alpha_2^{new,unc}\kappa &= y_2 - y_1 + f(\mathbf{x}_1) - \sum_{j=1}^{2}\alpha_j K_{1j} + \gamma K_{11} \\ &\quad -f(\mathbf{x}_2) + \sum_{j=1}^{2}\alpha_j K_{2j} - \gamma K_{12} - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1)) \\ &= y_2 - y_1 + f(\mathbf{x}_1) - f(\mathbf{x}_2) \\ &\quad +\alpha_2 K_{11} - \alpha_2 K_{12} + \alpha_2 K_{22} - \alpha_2 K_{12} - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1)) \\ &= \alpha_2\kappa + (f(\mathbf{x}_1) - y_1) - (f(\mathbf{x}_2) - y_2) - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1)) \end{aligned}$$

给出：

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{E_1 - E_2 - \varepsilon(\operatorname{sgn}(\alpha_2) - \operatorname{sgn}(\alpha_1))}{\kappa}$$

最后，必要情况下剪辑$\alpha_2^{new,unc}$确保其在区间$[U,V]$内。

7.6 高斯过程的实现技术

训练集：

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

上的贝叶斯学习问题，可以使用第6.2.3节中的高斯过程求解，高斯过程显示出与使

用协方差矩阵作为核的岭回归算法等价。求解这个问题包括计算 [见方程（6.10）] 满足下式的参数向量 α：

$$f(\mathbf{x}) = \mathbf{y}'(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k} = \alpha'\mathbf{k} = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

此处：

$$\begin{aligned} \alpha &= (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} \text{ 或} \\ \mathbf{y} &= (\mathbf{K} + \sigma^2\mathbf{I})\alpha \end{aligned}$$

这里 $\mathbf{k}$ 是一个向量，它是第 i 项为 $K(\mathbf{x}_i, \mathbf{x})$ 的向量，$\mathbf{K}$ 是核矩阵，项为 $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$。因此，问题的解可以通过求解 $\ell \times \ell$ 上的线性方程系统获得，而新的输入点上求解需要计算包括新点和每一个训练样例的核的值在内的内积。注意高斯过程情况下，参数向量 α 不是稀疏的。

已经有几种方法求解线性方程的系统，比如 LU 分解、Gauss Jordan 约减和对称矩阵的 Cholesky 分解。多数马上可以使用的数值软件包都提供了一些方法可供选择。遗憾的是求解一个线性方程系统的复杂性是 ℓ^3，使得这些方法不能用于大的数据集。

这个问题可以通过一个简单的梯度下降方法解决。事实上，第 2 章的练习 1 曾要求读者重写 Widrow-Hoff 算法的对偶表示形式。如果它可以没有偏置，并使用扩充的核 $\mathbf{K} + \sigma^2\mathbf{I}$ 来实现，就会收敛到高斯过程的解。处理这个计算复杂性问题的更高级的形式是共轭梯度技术。表 7.3 显示了共轭梯度算法的计算步骤。

表 7.3　共轭梯度方法的伪码

$\alpha^1 \leftarrow 0, \mathbf{h}^1 \leftarrow \mathbf{g}^1 \leftarrow \mathbf{y}$
for $k = 1, \ldots, n$
　　$\lambda \leftarrow \dfrac{(\mathbf{g}^k)'\mathbf{g}^k}{(\mathbf{g}^k)'\mathbf{C}\mathbf{h}^k}$
　　$\alpha^{k+1} \leftarrow \alpha^k + \lambda\mathbf{h}^k$
　　$\mathbf{g}^{k+1} \leftarrow \mathbf{g}^k - \lambda\mathbf{C}\mathbf{h}^k$
　　$\gamma \leftarrow \dfrac{(\mathbf{g}^{k+1})'\mathbf{g}^{k+1}}{(\mathbf{g}^k)'\mathbf{g}^k}$
　　$\mathbf{h}^{k+1} \leftarrow \mathbf{g}^{k+1} + \gamma\mathbf{h}^k$
end for
返回 α

这个过程保证在 $n = \ell$ 次迭代中收敛到解，但早期停止可以达到 $\alpha_{\max}$ 的一个逼近，它的复杂度仅为 $n \times \ell^2$。逼近的量可以使用 Skilling 方法估计，因此该方法为迭代算法给出了一个好的停止条件（更多细节见第 7.8 节给出的链接）。

7.7 习题

1. 为最大间隔优化问题实现一个梯度下降算法。在任意创造的数据集上测试。打印出 Karush-Kuhn-Tucker 条件。画出以迭代次数为函数的间隔。引入一些样例选择的启发式方法并重画。
2. 在不可分数据上尝试相同的算法。在算法中增加软间隔特征。试验可调参数。在小的真实数据上运行上面的算法，数据集可以从 UCI 数据库[96]获得。
3. 实现 SMO 分类算法，并在不同的数据集上运行。

7.8 补充读物和高级主题

凸优化问题从 20 世纪 50 年代起开始广泛研究，已经有很多技术可供使用。本章没有全部列出，仅简单地提供给读者易于实现的和典型的标准技术。优化算法的讨论，可以在[41,80,11,86]中找到。

对于支持向量机的特定情况，较好的综述有 Smola 和 Schölkopf [145]以及 Burges [23]。有关实现的问题和技术则包括 Kauffman[70]、Joachims[68]、Platt[114,112]、Osuna 和 Girosi[111] 以及 Keerthy 等人 [73,74]的讨论。

梯度上升算法引入到 SVM 中的论文是关于核 Adatron 的文章[44,24]；类似的想法独立出现于 Haussler 和 Jaakkola[65]的工作中。它们与固定 b 的 SMO[112] 及 Hildreth 的 QP 方法[62]相关。Mangasarian 和他的合作者最近提出了处理大规模数据集的算法[90,91,20]。尽管动机不同，Mangasarian 和 Musicant [89] 的连续过度松弛算法（SOR, Successive Over Relaxation）等价于第7.2节描述的随机梯度上升算法与 Platt[112]提出的样本启发式选择方法的结合。Mangasarian 和 Musicant[89]也使用 SOR 的连接给出了算法线性收敛的证明。

评注 7.4 讨论了使用固定偏置的可能，注意 Jaakkola 和 Haussler[65]也使用了这样的假设；Mangasarian 和 Musicant[89]讨论了这样的假设不大严格的若干实例。[170,45]中也使用了类似的方法。

梯度上升算法泛化性的在线理论由 Littlestone、Warmuth 和其他人 [75]提出。平方损失算法的讨论在[75]中做了介绍，而铰链损失的理论在[48]中介绍。这个理论提供了在线算法最坏情况下错误的最大数目的一个严格界，并且需要很少的假设。这个界也可以转换到由特定分布产生的数据的情况下。最有名的一个结果是产生了乘法更新算法，并且在这种情况下可以得到比标准梯度下降更好的结果。Cristianini 等人[29]研究过支持向量机的乘法算法。

Platt[112]设计了一流的 SMO 算法，并应用到文本分类问题中。本书的附录 A 给

出了 SMO 的伪码（John Platt 友情提供）。对 SMO 的偏置计算方法的一项改进在[74]中提出，并证实得到了更快的速度。Alex Smola 将 SMO 的算法推广到回归情况下[148,145]，代码可以在 GMD 的网站上找到[53]。

Keerthy 等[73]提供了一个非常上乘的 SVM 算法，它不需要通过最小权重向量的范数来最大化间隔。相反，他们注意到正负样本凸处的最近点的距离向量惟一决定了最大间隔超平面。问题的几何角度在[14,129]中讨论过。第 5 章的习题 2 要求读者为这个条件设计一个优化算法，因此促进为最大间隔问题提出不同解的策略。在相同的方法上，Kowalczyk[76]给出了可以用于硬软间隔问题的新的迭代算法的学习率，并给出了与其他迭代算法的实验比较，这些算法包括 Guyon 和 Stork[56]给出的软间隔优化迭代算法的变体。

支持向量机中的块技术早已经由 Vapnik 和 Chervonenkis 开始使用，很多论文涉及对其改进、推广和讨论，比如 Osuna 和 Girosi [111,110,109]、Joachims[68]、Platt[112]、Smola 和 Schölkopf [145] 以及 Kauffman[70]。Osuna 和 Girosi 的工作促进了后来者在数据选择上的工作，最终引出了 SMO 这样的算法。

参数自动微调的技术已经存在。比如[31]可以调整核参数，而ν-SVM允许用户设置用于分类[135]和回归[130]的支持向量个数的上界。可行间隙的停止条件在[148]中讨论，其中提出了一个类似方程（7.3）的条件。其他条件在[73,74]中进行了讨论。

在[50]中讨论了高斯过程的实现。核 Widrow-Hoff 的实验研究可以在[46]中找到。

著作[100]讨论了最优化问题的通用技术，并给出了商业优化软件包的链接。SVM 的早期实现就是基于这些优化软件包，比如 MINOS [101]、LOQO [156]、MATLAB [92]和其他上面提到的优化软件包。书[149]的第 1 章有不同实现方法的很好的概述。

SVM 实现的特定软件包可以在线获得。由 Royal Holloway、ATT 和 GMD FIRST 研究组准备的软件包可以在 London 大学 Royal Holloway 分校的网站[126]上获得。Joachims[68]的 SVM^{light} 也可以在网站[30]上获得。Alex Smola 准备的 SVM 分类的软件包可以在 GMD-FIRST 网站[53]上获得。这些软件包和其他软件包都可以在网站[30]中找到。

共轭梯度算法的原始文献是[61]，而在[49]中包含估计逼近质量的 Skilling 方法的讨论。高斯过程的软件也可在线获得，比如 Radford Neal 的代码[103]；Gibbs 和 MacKay 的代码[49]。

这些参考文献也在网站 www.support-vector.net 上给出，这个网站将不断及时补充新的研究成果并提供在线软件和论文的链接。

第8章　支持向量机的应用

本章将通过几个有趣的例子，展示书中所描述的算法的应用。支持向量机（SVM）已经应用到许多现实世界的问题中，因此本章的材料绝没有穷尽。本章主要是显示算法如何在很广泛的领域中成功的应用，以及在每种不同情况下如何做必要的选择。

将SVM应用到特定的实际问题需要解决大量的设计问题。本书没有详述应用领域的详情和如何获取训练数据等问题。并将这些视做已经确定的，在实践中它们会在SVM系统设计者和专门领域工作者的交互过程中得到很好的解决。

这样的合作或许也需要解决首要的设计问题，即为给定的应用选择适当的核。有几个标准选择，比如高斯的或多项式的核函数这些默认的选项，但如果它们被证明是无效的或者输入是离散结构，就需要精心制作的核了。通过隐式定义一个特征空间，核为机器检阅数据提供了一种描述语言。通常设计者在输入空间会面对相似的符号工作，这时领域专家能够给予宝贵的协助来帮助他们建立恰当的相似性度量。

一般而言，核的选择实际上是选择一族具有几个超参数的核，比如高斯核中参数σ需要确定。或许可以启发式选择这些参数：在高斯的情况下σ的一个好的选择是取不同类最近点的距离；通常这个选择必须适用于具体数据。缺乏可靠的条件时，在应用中需要使用验证集或者是交叉验证来设置这些参数。

SVM系统设计者面对的下一个决定是实现哪种SVM。假定数据需要分类，要决定是否使用最大间隔，或者选择哪种软间隔方法。这里的关键因素是数据的维数和噪声的类型及程度。

一旦确定核和优化条件，系统的关键部件都就位了。使用其他学习系统的经验可能会让你觉得，将进行一系列冗长的实验，参数不断变化直到达到满意的性能。本章报告的例子将演示最直接的SVM的实现，不需要进一步的修正，就能超出其他技术的能力。使用SVM的经验告诉我们，SVM可以成功应用的领域远远超出现在已经开发的领域。

本章描述的应用目的是提供几个SVM所能成功应用的例子。给出每个方面主要的研究情况，并提供相关的论文给那些想获得结果和方法细节描述的读者。本章重点描述给定应用中的问题和特定的核以及所使用的优化类型。

本章的例子包括文本（超文本）分类、图像分类、生物序列分析和生物数据挖掘、手写字符识别等。本书选择了这些领域中经常使用的简单例子，并在第8.5节给

出同样问题更复杂的研究的参考文献。

8.1　文本分类

文本分类的任务是将自然文本（超文本）文件根据内容分为预先定义的几个类别。很多领域都有这种问题，包括邮件过滤、网页搜索、办公自动化、主题索引和新闻故事的分类。因为一个文件可以分给不止一个类别，所以尽管这不是一个多分类问题，但可以视做一系列两分类问题，分别属于各类。

用于信息检索（IR，information retrieval）的文本的一种标准表示形式为构造 Mercer 核提供了一个理想的特征映射。因此，在其他领域得到的先验知识启发了核的设计，下面的各节会发现在其他许多情况下这是一个重复的模式。事实上，核以某种方式结合了距离的相似性度量。因此可以假定特定应用领域的专家已经发现了一些合理的相似性度量，尤其是在信息检索和生成模型领域。

8.1.1　IR 核应用于信息过滤

在信息检索文献中，一种寻找文本文件的通用技术是向量空间模型，也称为词袋表示。在这种方法中，一个文件 x 表示为预先固定集或术语词典索引的一个向量 $\phi(x)$；项可以是一个布尔变量，它标示对应术语是否存在，或以下面要描述的一种方式标示文件中术语的权重。最后，向量归一化，去除了文本长度的信息。因此，两个文件的距离可以通过计算相应向量的内积得到。在这个模型中，所有关于词的数量级的信息丢失了，类似第 8.2 节讨论的图像分类的情况。

信息检索的研究显示词干提供了很好的表示单元。一个词干是去除大小写和变形信息的词。比如单词 “computer”，“computation” 和 “computing” 都有相同的词干 “comput”。可以将数据库中的所有词干按照互信息排列，并去除无信息单词（比如 “and” 等）或称为停止单词然后选择处于前面的那些单词词干作为字典。

每个词干的权重因子的标准选择如下：

$$\phi_i(x) = \frac{\mathrm{tf}_i \log(\mathrm{idf}_i)}{\kappa}$$

这里 tf_i 是术语 i 在文件 x 中发生的次数，idf_i 是所有文件的数目与包含这个术语的文件数目的比，κ 是归一化的常数确保 $\|\phi\|_2 = 1$。

按这种方式，文件由向量表示，向量的项记录了文件中特定词干使用的次数，并根据词干的信息量加权。一般 ϕ 有几万项，通常跟训练的样本数目相当或要多。进一步，对特定的文件，表示通常是非常稀疏的，只有少数非零项。既然算法和评

价函数仅需要计算数据的内积，可以采用一系列技术来加快计算。

函数 $K(x,z)=\langle\phi(x)\cdot\phi(z)\rangle$ 明显是一个合理的核，因为它是显式构造的特征空间的内积。它的核矩阵因而总是半正定的，SVM 可以用 $K(x,z)$ 来判别。由于极端稀疏性和向量 $\phi(x)$的高维特性，假定特征空间中的点易于分开是合理的，因此一个标准硬间隔 SVM 即足够。进一步为计算稀疏内积设计快速技术是可能的。

Joachims[67]和 Dumais 等人[36]选择的数据集是标记的路透社新闻故事，即 Reuters- 21578 数据集，它是 David Lewis 从 1987 年路透社新闻专线公开发表的数据中编辑的。其中，采用了一种数据的特定分开方法，称为“ModApte”，包括 9603 个文件的训练集和 3299 个文件的测试集，并分为 90 个类别。经过单词词干预处理后，去除停止单词，这个文集包括 9947 个不同的术语，这些术语至少在 3 个文件中出现过。故事平均长度为 200 个单词。数据集类别的例子是社团公告、收入、货币市场、谷、麦、船等，每一个有不同数目的例子。注意许多文件同时分给了几个类别。

Joachims 所做的更多的实验是在 Oshumed（Oregon 保健科学大学）文集上，是 William Hersh 在 1991 年编辑的，包括 50216 个带摘要的文件，前 10000 个文件用来做训练，接下来的 10000 个文件用来做测试。每个文件赋予表示疾病的 23 个类别中的一个或多个。预处理（词干和去除停止单词）后，训练集包括 15 561 个至少在 3 个文件中出现过的不同的术语。

在 Joachims 所做的实验中，使用了一个简单的最大间隔 SVM，并启发式去除了那些α_i过大的文件，将其视做离群点。这个策略能够替代软间隔技术的需求，因为数据使用了特定的表示。低阶的多项式核和高斯核得到了很好的结果。独立选择核参数，SVM 表现出比标准方法如简单贝叶斯、Rocchio、决策树算法 C4.5 和 k 近邻算法更好的性能。

评注 8.1 这些实验中所用的训练算法是 $\text{SVM}^{\text{light}}$，由 Joachim 开发，并可以从因特网上获得（见第 7.8 节）。

评注 8.2 有趣的问题是，为什么 SVM 在这个应用中特别有效。在传统的系统中，当特征的数目比训练数目大时，可能会出现第 4 章描述的线性函数在高维空间泛化能力差的情况。要解决这个问题，学习系统只能充分利用目标函数和分布之间的良好联系。SVM 间隔最大化优化了这个优势，因此克服了高维表示中的困难。

评注 8.3 这个方法的特征通过计算文件的词频获得，是低等级的。它们仅捕捉文件的全局特性。仍然需要完成在更高级的特征上使用更复杂的核的工作。第一步就是要考虑在第 3 章例 3.15 介绍的字符串核。这个核不考虑词或子字符串出现的次数，因此不能选择要包括哪个词，但是它可以在更高维中工作，这是特征向量不再显式计算带来的可能。

8.2　图像识别

大批量的数字图像容易从因特网或者是某个数据库获得。而且，图像的产生越来越廉价，已在很多应用中使用。图像的自动分类在许多应用领域都是一项关键任务，这些领域包括信息检索、因特网数据过滤、医学应用、可视场景的目标检测等。图像分类中的研究多数集中于图像高层次特征的提取，比如使用边缘检测技术和形状描述技术来获取图像的相关属性而不增加过多特征。然而，只采用低层次特征的快速信息检索技术也已经得到研究。

当直接在图像上操作时，传统的分类方法由于数据的高维特性表现较差，但是 SVM 可以克服极高维表示的缺陷，这一点在上一节的文本分类的例子中已被证明。一个类似文本分类技术的方法同样用于图像分类任务，并且在这种情况下，线性硬间隔学习器通常具有很好的泛化能力。在图像检索任务中，通常使用灰度或颜色直方图这样的低层次特征。它们捕捉了图像的全局低层次特性，概念上同文本分类例子中使用的相似。像词干那样，直方图也不能保留位置信息，直方图之间的距离则可用χ^2或者其他描述分布间差异的测度来估计。

在其他情况下，图像可以直接用位图矩阵表示，这也是第 8.3 节描述的手写数字识别的表示方法。现在开始探讨用这种表示做实验。

仍需要探索使用更复杂的核，以便得到更好的结果。与实践者的交互应有助于设计这样的核。

8.2.1　视位无关分类

Pontil 和 Verri[118]对这些条件下 SVM 的性能做了研究，使用 SVM 做与视位无关的目标识别。他们使用位矩阵或者位图矩阵这样最简单的方式来表示图像。矩阵视做向量输入。如果矩阵有高 h 和宽 w，彩色图像成为长度为 $3 \times h \times w$ 的向量，而灰度图像的向量长度为 $h \times w$。向量的每个分量是特定位置的像素。

基准测试基于 Columbia 目标图像库（COIL, Columbia Object Image Library，从因特网获得）数据集，包括 7200 幅图像：100 个不同三维目标的 72 个不同视角，每一个集的 72 个视角对应着相同目标的 5 度旋转。它们转换为灰度图片，通过 4×4 小块取平均解析率从 128×128 降到 32×32。这个前处理阶段完成后，数据集的每个图像转换成一个 1024 维的向量，每个灰度用 8 比特的数表示。选择这些向量的一个简单线性内积作为核，并实现了基本的最大间隔分类器。

注意系统没有进行高层次特征的提取，而是将图像视为高维像素空间的点来分类。7200 幅图像分为两组，3600 幅做训练组，其他做测试组，每组包括 100 幅图像

10 度旋转的一个子集。这是个多类问题，可以通过为每两类训练一个分类器然后以某种方式组合（详见论文）来解决。对每个实验，仅使用了从 100 个类中随机选择包含的 32 类的子集。系统在三维目标的 36 个视角上训练，然后用其余 36 个视角测试。

尽管特征描述具有高维性，采用不含核的硬间隔超平面仍足以将这些数据准确分开。这也足够在测试集上达到极高的性能。在许多情况下，只有在数据上附加人工噪声才能使学习器误预报。为了比较，在相同条件下训练的标准感知机在性能上表现较差。

8.2.2 基于颜色的分类

实验[118]全部依赖灰度等级的信息，忽略了像颜色这样重要的信息来源。这个信息可以用于基于 SVM 的分类，就像在 Olivier Chapelle 及其合作者[25]做的一系列实验一样，他们仅使用了颜色和光照信息。

特征使用的方法有些类似文本分类一节中的使用。基于已建立的信息检索技术，他们使用了一种图像的相似性度量，就是图像各自直方图的距离。遗憾的是，他们没有证明这样的核满足 Mercer 条件。然而有趣的是，这是一个利用领域专家知识构造核，然后应用到 SVM 中的例子。事实上，相同的模式可以也在生物序列的例子中看到。

选择了一种图像的描述方式，也就指定了核。从技术上讲，就是选择一个从图像到特征向量的映射中，和一个特征向量上的度量。就像上面讨论的，描述图像最简单的方式是位矩阵或位图。这种表示不具有尺度和平移不变性。一种稍微复杂的表示是利用颜色的直方图。每种颜色是三维颜色空间的一个点。每个图像与一个直方图相关联，它编码了每种颜色像素的比率。这种情况下的特征是一组颜色区或颜色柱，特征空间的维数取决于柱子的大小。这种表示相对其他操作的一个明显优势是具有不变性，可以比较大小不同的图像。在[25]中，这与色度、饱和度、明度（HSV, Hue Saturation Value）结合，而不是红、绿、蓝（RGB）结合。这两种表示是等价的，但是前者将颜色分量（HS）和明度分量（V）分离开来，因此认为感知上是更合理的。

要指定核函数，必须要确定特征向量之间的相似性度量。一类通用的核可以写做：

$$K(\mathbf{x},\mathbf{z}) = \exp\left(-\frac{d(\mathbf{x},\mathbf{z})}{\sigma^2}\right)$$

这里 d 是输入的相似性的某种度量。对于直方图，一个可能的选择是 χ^2 函数，由下

式逼近：

$$d(\mathbf{x},\mathbf{z})=\sum_{i=1}^{n}\frac{(x_i-z_i)^2}{x_i+z_i}$$

另一个选择是简单的 p-范数：

$$d_p(\mathbf{x},\mathbf{z})=\left(\sum_{i=1}^{n}|x_i-z_i|^p\right)^{1/p}$$

当 $p=1,2$ 时，使用 d_p 获得的核满足 Mercer 条件。对于 χ^2 还不能确定，但是这不影响其直觉理解和在实践中的良好表现。此外，也可以从图像中提取其他类型的直方图用做特征向量。

为了说明这方面的应用，执行了一项图像识别任务，其中所用的图像是从 Corel Stock 图像集提取的。这个 Corel 数据集包括了大约 200 个类别，每个类别包括 100 幅图像。在实验中，Corel14 表示的子集是特别选择包含与下面 14 个不同类别有关的图片：飞机展、熊、大象、虎、阿拉伯马、北极熊、非洲特有动物、猎豹、美洲狮、秃鹰、山岭、沙漠、日出及夜景。在这个简化的子集中，有 1400 幅图像。每个类别的 2/3 用来训练，1/3 用来测试。每维上的颜色柱固定为 16 个，这样直方图的维数就是 16^3=4096，所用的核就是上面叙述的那些，并且采用 1-范数，2-范数和 χ^2 作为相似性度量。

结果显示在相同的度量标准上比 k 近邻算法提高了两倍。有趣的是，三个矩阵中，一阶和 χ^2 表现相似，而二阶表现很差。因此，对这个问题，标准的高斯核不是一个好的选择。

评注 8.4　注意，在两种情况下，样例的数目都是小于特征空间维数的。但尽管这样，最大间隔分类器仍然能获得好的性能。此外，使用最大间隔分类器的可能性部分来自特征空间的高维特性。

评注 8.5　在可视场景中的目标检测这样一项很重要的应用中，SVM 被证明特别有效。一个例子是人脸检测：给定任意图像作为输入，检测其中是否有人脸存在，以及脸的位置在何处。系统是 Osuna 等人[110]开发的，尽可能扫描像人脸的模式，然后利用 SVM 作为分类器，检查一幅给定的图像是否是人脸。数据库包含脸和非脸模式，图像用19 × 19 = 361个像素的向量表示，训练一个软间隔分类器，使用二阶多项式核。在应用中考虑了大量的技术细节，不可能在这里详述。一个相似的问题是在汽车前进的可视场景中检测行人的应用[108]，这里小波作为特征提取的方法，在输入多项式核 SVM 之前做了前处理。这些应用都很重要，但不可能在这里详细描述。

请看第 8.5 节提供的原始论文的链接。

8.3 手写数字识别

支持向量机在现实世界应用的第一个例子是手写字符识别问题。这个问题通常用做分类器的测试平台，最初是为满足美国邮政服务局使用手写邮政编码自动分类邮件的需要提出的。SVM 的不同模型在两个数字集上做过测试，这两个数字集分别由 USPS（美国邮政服务局）和 NIST（国家标准技术局）公开提供。USPS 数据集包括 7291 个训练样本，2007 个测试样本，用 256 维的向量（16×16矩阵）表示，每个点的灰度值从 0 到 255。NIST 数据集是为了做测试平台用的，包括 60 000 个训练样本和 10000个测试样本，图像用20×20的矩阵表示，其中各项也是灰度值。

Vapnik 和他的合作者[19,27,128,78]处理过这个问题，用的都是最大间隔和软间隔分类器，主要是用高斯和多项式核，尽管 S 形核不符合 Mercer 条件，也使用过。进一步，多类的 SVM 也在这些数据上测试过。有趣的是，实验不光将 SVM 跟其他分类器做比较，还对不同类型的 SVM 做了比较。它们显示出近似的性能，因此使用几乎相同的支持向量，它们独立于所选择的核。如所预料的，当核映射的能力上升，支持向量的数目也增加了，但这发生得很慢。核参数在一定范围内变化时，性能基本稳定。实验结果在 Burges，Cortes，Schölkopf，Vapnik 等学者的一些系列论文中报告过，在[159]中做了总结。

对 USPS 数据，输入空间是 256 维，对不同的d和σ值，使用了下列多项式和高斯核：

$$K(\mathbf{x},\mathbf{y}) = \left(\frac{\langle \mathbf{x} \cdot \mathbf{y} \rangle}{256}\right)^d$$

和

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{256\sigma^2}\right)$$

对于多项式核，阶数从 1 到 6 都测试过，对于高斯核σ的值在 0.1 到 4.0 之间。核参数肯定影响泛化能力。USPS 可以使用三阶多项式核的最大间隔分类器完全分开，而使用一阶软间隔优化的情况下，一阶多项式核的训练误差是340/7291，二阶的训练误差是4/7291。支持向量的数目随阶数增加很慢。对高斯核，可以使用自适应核算法[31]，结合第 7.2 节描述的梯度上升算法，自动扫描可能核参数的空间。

这套实验特别有趣，因为这些数据被广泛研究过，还有专门为这个数据集定制的算法，它结合了问题的大量先验知识。SVM 没有包含任何细节的先验知识就能与这些算法的效果一样好，这是很显著的成就[78]。

8.4　生物信息学

8.4.1　蛋白质同源检测

20 个氨基酸序列形成蛋白质，并且存在成千上万种不同的蛋白质，生物信息学的一个核心问题是在氨基酸序列基础上预报一个蛋白质的结构和功能特征。这可以通过将新的蛋白质序列关联到性质已知的蛋白质序列，比如检测蛋白质的同源性。在某种意义上，蛋白质是按族聚类的，它们根据各自的性质组成超类。

存在很多技术基于蛋白质的序列检测它们之间的同源性。一种常用的技术是从正样本中构造一个蛋白质族的广义模型，使用它计算新的序列与这族之间的相似程度。一个统计模型，比如隐马尔可夫模型（HMM, Hidden Markov Model）中 $H(\theta)$的参数 θ 是通过将样例跟蛋白质性质的先验知识相结合估计得到的。

模型 $H(\theta)$赋予新的蛋白质序列 x 一个概率 $P(x|H(\theta))$，从相同超族来的蛋白质较之非本族的蛋白质将有更高的分数。这个概率分数是蛋白质和超族的同源性的度量。可以看出广义模型也可以开发成为一个核。Jaakkola 和 Haussler[65]的实验目的是研究这样的核构造的 SVM 能否有效地将蛋白质族分类到它们的超族中。

核函数确定一对序列之间的相似性分数，而 HMM 度量的是序列同模型的相似性分数。然而 HMM 系统将新的蛋白质排序的过程中会有一些中间量，可用于实现到特征空间的映射，也就是核。在 HMM 的内部语言中，它们可以看做蛋白质的中间表示。这里不再详细描述 HMM，中间量中包含一个 Fisher 分数，对一个序列来说就是：

$$U_x = \frac{\partial \log P(x|H(\theta))}{\partial \theta}$$

它是序列 x 对应于模型$H(\theta)$的参数的 log 相似性的梯度。向量U_x 可以看做 HMM 中所查询序列的中间表示：这个表示反映了构造 HMM 的假设，即生物学先验知识。它与概率模型统计的向量密切相关，这个模型用中间语言给出了序列的完整总结，向量可以作为在给定序列上评价 HMM 时容易计算的一个副产品。

很自然，两个向量的距离提供了两个对应序列的相似性度量。因此可以在二阶范数上利用高斯函数构造核：

$$K(x,y) = \exp\left(-\frac{\|U_x - U_y\|_2^2}{2\sigma^2}\right)$$

尽管更自然的是使用 Fisher 信息矩阵提供的度量。核 K 是特征空间的内积，这个空间是两个特征映射组合的映像：

$$x \longmapsto U_x \longmapsto \phi(U_x)$$

这里ϕ是与高斯相关的特征映射。因此它肯定是一个 Mercer 核。

系统是使用类似第 7.2 节描述的梯度上升优化实现的。正训练样本是一个超族，排除了某个已知家族的所有成员，后者成为测试样本。测试和训练集中的负样例是从其他超族得到的。

系统明显比当前流行的一些蛋白质同源检测系统要好，尤其是在检测较远的同源时更有效果。

评注 8.6 值得强调的是核的构造使用了编码入 HMM 的领域知识。许多其他的生物序列之间距离的度量也已经提出，包括使用编辑距离、广义语法、HMM 对等。它们的目的都是将领域先验知识融入到输入的相似性度量中。

8.4.2 基因表达

SVM 在生物数据挖掘中的另一个应用是从 DNA 芯片中得到的基因表达数据的自动分类。DNA 芯片技术是革命性的基因分析，这使得可以检测表达特定组织的基因和比较不同条件下组织的基因表达的等级成为可能。生物学家可以在一个实验中很容易测定成千上万个基因表达的等级。当 DNA 芯片产生的数据不断积累，越来越有必要准确提取生物信息，最终自动指出基因的功能。作为例子，可能喜欢使用这样的数据来决定一个新的基因是否编码了特定类的蛋白质。这样的分类器可以为一类蛋白质从未知功能的基因中识别新的成员。这种分类任务对于传统模式识别系统是很困难的，因为数据集很大，有噪声且是不平衡的，负样例远远多于正样例。

Brown 等人[21]描述了一个成功将 SVM 应用到基因表达数据中的例子，它既要分开未知的基因，又要找到已知干净的数据集中同等重要的一个。这里不再描述产生芯片数据的有趣的实验准备工作。Brown 等人的实验使用了下面的数据集做计算机分析。发芽酵母 *S. cerevisiae* 的基因总共 2467 个，用 79 维基因表达向量表示，并根据功能分为 6 个类别。多数情况下，正的数目少于 20 个（少于 1%），而且噪声的存在使得学习任务异常艰难：所有训练数据都被分为负的平凡解是经常发生的。

使用的 SVM 是第 6 章描述的二阶软间隔优化的变体，其中在核矩阵增加一个对角元素。在这种情况下，为了分别控制两类错误数，根据类别在核矩阵中加入对角元素。这样设计是为了在占统治地位的负样例上产生小的拉格朗日乘子，在正样例上产生大的拉格朗日乘子。两类元素的比率固定到大约等于两类样本数目的比率。选定的核是高斯分布的，σ固定大体等于最近的两对正样例的距离。

获得的结果同 Parzen 窗算法、Fisher 线性判别和决策树算法 C4.5 相比，SVM 在所有的测试集上都要好，多数情况下优越因子为 2。系统也用于离群点的检测和生物

数据的数据浏览器[22]的核心引擎。

8.5　补充读物和高级主题

本章描述的这些应用仅仅是最近几年 SVM 各种各样应用的几个例子。本书选择这些特定例子的原因是它们容易获得，或者是展示了学习系统设计的某个重要方面。没有描述其他重要的例子是因为篇幅有限或者是它们太复杂，这些示例和最近的其他一些应用都可从网站[30]获得。

设计中的关键是核的选择。创立好的核需要多方面的思考：输入之间相似性的许多度量已经在一些相关问题中研究过，要知道它们中的哪一个能够成为好的核，需要具有对应用领域的深入洞察能力。这里描述的一些应用都使用了很简单的核，这由信息检索技术启发，比如计算机视觉[129,17,118,25]中使用的，文本分类[36,67]中使用的。即使使用如此简单的核，而且特征个数远远大于输入的样本个数，SVM 仍然得到了相当准确的假设，通常比其他标准算法要好。其他应用使用了很复杂的核，比如生物序列[65]中用到的，其中核由整个 HMM 系统提供。Watkins 和 Haussler 最近为符号序列（在第 3.7 节讨论）提供了优雅的核，在问题中的应用中与生物序列分析和文本分类一样，得到了很有趣的结果。

在生物数据挖掘的例子[21]中，使用的核是一个简单的高斯核，但是困难在于数据集不平衡，这个问题可以使用类似二阶软间隔优化来解决。相同的技术用在其他的医学应用中：SVM 用于 TBC 的自动诊断[168]来控制假设的特定性和敏感性。基因表达的工作也使用了 SVM 的其他重要特征：用于数据清洗[55]，这里潜在的离群点从已经得到的支持向量中寻找。数据和实验可以从网站[22]获得。

其他计算机视觉的应用包括人脸检测和行人检测，研究者有 Tomaso Poggio、Federico Girosi 和他们的合作者[108,110]，还包括结核诊断[168]，目标识别的实验[129]。更多 SVM 的应用可以查询两本文集[132,149]和 GMD-FIRST 的网站[53]或其他可以获取的网站[30]。

系统参数的调节是另一个重要的设计主题。启发式是本章介绍的一个例子，这是 Jaakkola 研究的高斯核 σ 的选择；一般来说可以在一些学习原理的基础上考虑参数自动适应数据的方案。这样的一个简单方案在[31]中给出。

这些参考文献也在网站 www.support-vector.net 上给出，这个网站将不断及时补充新的研究成果并提供在线软件和论文的链接。

附录 A　SMO 算法的伪码

```
target = desired output vector
point = training point matrix

procedure takeStep(i1,i2)
  if (i1 == i2) return 0
  alph1 = Lagrange multiplier for i1
  y1 = target[i1]
  E1 = SVM output on point[i1] - y1 (check in error cache)
  s = y1*y2
  Compute L, H
  if (L == H)
    return 0
  k11 = kernel(point[i1],point[i1])
  k12 = kernel(point[i1],point[i2])
  k22 = kernel(point[i2],point[i2])
  eta = 2*k12-k11-k22
  if (eta < 0)
  {
    a2 = alph2 - y2*(E1-E2)/eta
    if (a2 < L) a2 = L
    else if (a2 > H) a2 = H
  }
  else
  {
    Lobj = objective function at a2=L
    Hobj = objective function at a2=H
    if (Lobj > Hobj+eps)
      a2 = L
    else if (Lobj < Hobj-eps)
      a2 = H
    else
      a2 = alph2
  }
  if (|a2-alph2| < eps*(a2+alph2+eps))
```

```
    return 0
  a1 = alph1+s*(alph2-a2)
  Update threshold to reflect change in Lagrange multipliers
  Update weight vector to reflect change in a1 & a2, if linear SVM
  Update error cache using new Lagrange multipliers
  Store a1 in the alpha array
  Store a2 in the alpha array
  return 1
endprocedure

procedure examineExample(i2)
  y2 = target[i2]
  alph2 = Lagrange multiplier for i2
  E2 = SVM output on point[i2] - y2 (check in error cache)
  r2 = E2*y2
  if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0))
  {
    if (number of non-zero & non-C alpha > 1)
    {
      i1 = result of second choice heuristic
      if takeStep(i1,i2)
        return 1
    }
    loop over non-zero and non-C alpha, starting at random point
    {
      i1 = identity of current alpha
      if takeStep(i1,i2)
        return 1
    }
    loop over all possible i1, starting at a random point
    {
      i1 = loop variable
      if takeStep(i1,i2)
        return 1
    }
  }
  return 0
endprocedure

main routine:
  initialize alpha array to all zero
  initialize threshold to zero
  numChanged = 0
  examineAll = 1
  while (numChanged > 0 | examineAll)
    {
```

```
    numChanged = 0
    if (examineAll)
      loop I over all training examples
        numChanged += examineExample(I)
    else
      loop I over examples where alpha is not 0 & not C
        numChanged += examineExample(I)
    if (examineAll == 1)
      examineAll = 0
    else if (numChanged == 0)
      examineAll = 1
  }
```

附录B 背景数学

B.1 向量空间

先从向量空间的定义开始。确切地，这里只给出一个简单的定义，在损失一些一般性的情况下，仍然足以在本书使用。比如向量空间可以定义在任何域上，这里只考虑定义在实数域上，因此此处介绍的有时称为实数向量空间。

定义 B.1 如果在集合 X 上定义两个操作（加，标量乘），使得对 $\mathbf{x}, \mathbf{y} \in X$ 和 $\alpha \in \mathbb{R}$ 有：

$$\begin{aligned} \mathbf{x}+\mathbf{y} &\in X \\ \alpha\mathbf{x} &\in X \\ 1\mathbf{x} &= \mathbf{x} \\ 0\mathbf{x} &= \mathbf{0} \end{aligned}$$

还使得 X 在加操作下与单位矩阵 $\mathbf{0}$ 是交换群，并且标量乘满足分布律：

$$\alpha(\mathbf{x}+\mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y}$$

和

$$(\alpha+\beta)\mathbf{x} = \alpha\mathbf{x} + \beta\mathbf{x}$$

则集合 X 是一个向量空间（VS）。X 的元素称为向量，而实数称为标量。

例 B.2 VS 的一个标准样例是固定维数为 n 的实数列向量组成的集合 $\mathbb{R}^n$。使用撇号表示向量（和矩阵）的转置，使得一个一般的列向量可以写为：

$$\mathbf{x} = (x_1, \ldots, x_n)'$$

这里 $x_i \in \mathbb{R}, i = 1, \ldots, n$。

定义 B.3 如果限制向量空间 X 中非空子集 M 的操作使其成为一个向量空间，则 M 是 X 的子空间。

定义 B.4 向量空间中向量 $\mathbf{x}_1, \ldots, \mathbf{x}_n$ 的线性组合是加和形式 $\alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n$，这里 $\alpha_i \in \mathbb{R}$。如果 α_i 是正数，并且 $\sum_{i=1}^n \alpha_i = 1$，这个和又称为凸组合。

容易发现子空间中向量的线性组合仍然在子空间内，线性组合实际上可以用来

从 VS 的一个任意子集构造子空间。如果 S 是 X 的一个子集，用 span(S)表示 S 中向量的所有可能线性组合构成的子空间。

定义 B.5 对于向量的有限集合 $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$，如果能找到常数 $\alpha_1, \ldots, \alpha_n$，并且至少有一个非零，使得：

$$\alpha_1 \mathbf{x}_1 + \cdots + \alpha_n \mathbf{x}_n = \mathbf{0}$$

则称 S 为线性相关，否则称为线性无关。

这意味着若 S 是一个线性无关的向量集合，span(S)中的向量 $\mathbf{y}$ 对某个 n 和 $\mathbf{x}_i \in S, i = 1, \ldots, n$有惟一的表示形式：

$$\mathbf{y} = \alpha_1 \mathbf{x}_1 + \ldots + \alpha_n \mathbf{x}_n$$

定义 B.6 对向量集合 $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$，如果 S 是一个线性无关的集合，并且对每个 $\mathbf{x} \in X$能够惟一表示为 S 中向量的线性组合，则称 S 形成了 X 的基。尽管总是有许多不同的基，但它们的大小是相同的。向量空间中基的大小称为维数。

有限维空间易于研究，因为极少出现病态情况。有限维 VS 的基本性质扩展到无限维情况时需要附加特定要求。下面先介绍 VS 中距离的一个度量。

定义 B.7 线性赋范空间是向量空间 X 与实值函数的结合，这个函数将每个元素 $x \in X$映射到一个实数 $\|\mathbf{x}\|$，即 $\mathbf{x}$ 的范数，同时集合满足下列三个性质：

1. **正性** $\|\mathbf{x}\| \geqslant 0, \forall \mathbf{x} \in X$，当且仅当 $\mathbf{x} = \mathbf{0}$时等式成立；
2. **三角不等性** $\|\mathbf{x} + \mathbf{y}\| \leqslant \|\mathbf{x}\| + \|\mathbf{y}\|$，$\forall \mathbf{x}, \mathbf{y} \in X$；
3. **同质性** $\|\alpha \mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$，$\forall \alpha \in \mathbb{R}$和$\forall \mathbf{x} \in X$。

例 B.8 考虑实数的可数序列并且令 $1 \leqslant p < \infty$。空间 ℓ_p 是序列 $\mathbf{z} = \{z_1, z_2, \ldots, z_i, \ldots\}$的集合，并满足：

$$\|\mathbf{z}\|_p = \left(\sum_{i=1}^{\infty} |z_i|^p \right)^{1/p} < \infty$$

空间 ℓ_∞ 由序列 $\mathbf{z}$ 形成，并满足：

$$\|\mathbf{z}\|_\infty = \max_{i \in \mathbf{N}} \left(|z_i| \right) < \infty$$

向量 $\mathbf{x}$ 和 $\mathbf{y}$ 的距离可以定义为它们差的范数，即 $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$。

定义 B.9 在线性赋范空间中，对向量 $\mathbf{x}_n$，如果实数序列 $\|\mathbf{x} - \mathbf{x}_n\|$收敛到 0，则称 $\mathbf{x}_n$ 的无限序列收敛于向量 $\mathbf{x}$。

定义 B.10 线性赋范空间中的序列$\mathbf{x}_n$，如果当$n,m \to \infty$时，$\|\mathbf{x}_n - \mathbf{x}_m\| \to 0$，称$\mathbf{x}_n$为柯西序列。更准确地说，给定$\varepsilon > 0$，存在整数$N$使得对所有$n,m > N$有$\|\mathbf{x}_n - \mathbf{x}_m\| < \varepsilon$。对某个向量空间，当每个柯西序列都收敛到空间的一个元素，称该空间是完备的。

注意在赋范空间每个收敛序列都是柯西序列，但逆命题不成立。其中每个柯西序列都有一个极限的空间称为完备空间。完备线性赋范空间称为 Banach 空间。

B.2 内积空间

内积空间的理论是一个工具，可以用于几何、代数、积分、泛函分析和逼近理论。本书从不同程度对其加以使用，这里总结所用到的主要结论。再次强调，本书只讨论实数情况下的主要结论。

定义 B.11 对从向量空间X到向量空间Y的函数f，如果对所有$\alpha, \beta \in \mathbb{R}$并且$\mathbf{x}, \mathbf{y} \in X$满足：

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$$

则称函数f为线性函数。

注意，可以将实数视为维数为 1 的向量。因此，实值函数如果满足相同的定义也是线性的。

定义 B.12 令$X = \mathbb{R}^n$和$Y = \mathbb{R}^m$。从X到Y的线性函数可以表示成项为$\mathbf{A}_{ij}$的$m \times n$矩阵$\mathbf{A}$，使得向量$\mathbf{x} = (x_1, \ldots, x_n)'$映射到向量$\mathbf{y} = (y_1, \ldots, y_m)'$，这里：

$$y_i = \sum_{j=1}^{n} \mathbf{A}_{ij} x_j \quad i = 1, \ldots, m$$

对$i \neq j$，项$\mathbf{A}_{ij} = 0$的矩阵称为对角矩阵。

定义 B.13 对向量空间X，如果存在双线性映射（在每个参数上是线性的），对两个元素$\mathbf{x}, \mathbf{y} \in X$，可以给出一个用$\langle \mathbf{x} \cdot \mathbf{y} \rangle$表示的实数，并满足下列性质：

- $\langle \mathbf{x} \cdot \mathbf{y} \rangle = \langle \mathbf{y} \cdot \mathbf{x} \rangle$
- $\langle \mathbf{x} \cdot \mathbf{x} \rangle \geqslant 0$并且$\langle \mathbf{x} \cdot \mathbf{x} \rangle = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$

则X称为内积空间。

量$\langle \mathbf{x} \cdot \mathbf{y} \rangle$称为$x$和$y$的内积，有时称为点积或标量积。

定义 B.14 令$X = \mathbb{R}^n$，$\mathbf{x} = (x_1, \ldots, x_n)'$，$\mathbf{y} = (y_1, \ldots, y_n)'$。令$\lambda_i$为固定的正数。下面定义了一个有效内积：

$$\langle \mathbf{x} \cdot \mathbf{y} \rangle = \sum_{i=1}^{n} \lambda_i x_i y_i = \mathbf{x}' \Lambda \mathbf{y}$$

这里Λ是项$\Lambda_{ii} = \lambda_i$非零的$n \times n$对角矩阵。

定义 B.15 令$X = C[a,b]$是实数区间$[a,b]$上连续函数的向量空间，具有明显的加以及标量积操作。对$f, g \in X$，定义：

$$\langle f \cdot g \rangle = \int_a^b f(t)g(t)dt$$

从内积空间的定义，进一步的两个性质为：

- $\langle \mathbf{0} \cdot \mathbf{y} \rangle = 0$
- X自动成为赋范空间，范数为：

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x} \cdot \mathbf{x} \rangle}$$

定义 B.16 对X的两个元素$\mathbf{x}$和$\mathbf{y}$，如果$\langle \mathbf{x} \cdot \mathbf{y} \rangle = 0$，它们称为垂直的。对$X$中向量的集合$S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$，如果$\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle = \delta_{ij}$，称$S$为正交的，这里$\delta_{ij}$当$i = j$时为1，否则为0。对一个正交集$S$和向量$\mathbf{y} \in X$，表达式：

$$\sum_{i=1}^{n} \langle \mathbf{x}_i \cdot \mathbf{y} \rangle \mathbf{x}_i$$

称为$\mathbf{y}$的傅里叶序列。

如果S形成一个正交基，每一个向量$\mathbf{y}$等于它的傅里叶序列。

定理 B.17 （Schwarz不等式）在内积空间中：

$$|\langle \mathbf{x} \cdot \mathbf{y} \rangle|^2 \leqslant \langle \mathbf{x} \cdot \mathbf{x} \rangle \langle \mathbf{y} \cdot \mathbf{y} \rangle$$

当且仅当$\mathbf{x}$和$\mathbf{y}$相关时，等号成立。

定理 B.18 对内积空间X中的向量$\mathbf{x}$和$\mathbf{y}$，有：

$$\begin{aligned} \|\mathbf{x}+\mathbf{y}\|^2 &= \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 + 2\langle \mathbf{x} \cdot \mathbf{y} \rangle \\ \|\mathbf{x}-\mathbf{y}\|^2 &= \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\langle \mathbf{x} \cdot \mathbf{y} \rangle \end{aligned}$$

定义 B.19 内积空间中向量$\mathbf{x}$和$\mathbf{y}$的夹角θ定义为：

$$\cos\theta = \frac{\langle \mathbf{x} \cdot \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

如果$|\langle \mathbf{x} \cdot \mathbf{y} \rangle| = \|\mathbf{x}\| \|\mathbf{y}\|$，余弦为1，$\theta = 0$，$\mathbf{x}$和$\mathbf{y}$称为平行。如果$\langle \mathbf{x} \cdot \mathbf{y} \rangle = 0$，余弦为

0，$\theta = \frac{\pi}{2}$，称向量之间为垂直。

定义 B.20 给定内积空间 X 中的向量集 $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$，项 $\mathbf{G}_{ij} = \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ 的 $n \times n$ 矩阵 $\mathbf{G}$ 称为 S 的 Gram 矩阵。

B.3 希尔伯特空间

定义 B.21 对空间 H，如果存在可数子集 $D \subseteq H$，使得 H 中的每个元素是 D 中元素序列的极限，H 称为可分的。希尔伯特空间是完备可分内积空间。

有限维向量空间，比如 $\mathbb{R}^n$，是希尔伯特空间。

定义 B.22 令 H 是希尔伯特空间，M 是 H 的封闭子空间，并且 $\mathbf{x} \in H$。存在惟一向量 $\mathbf{m}_0 \in M$，称为 M 上 $\mathbf{x}$ 的投影，满足：

$$\|\mathbf{x} - \mathbf{m}_0\| \leqslant \inf\{\|\mathbf{x} - \mathbf{m}\| : \mathbf{m} \in M\}$$

$\mathbf{m}_0 \in M$ 成为 M 上 $\mathbf{x}$ 的投影的充分必要条件是向量 $\mathbf{x} - \mathbf{m}_0$ 与 M 中的向量是垂直的。

该定理的一个重要结果是在由正交向量 $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ 产生的子空间 M 中对 $\mathbf{x}$ 的最佳逼近由它的傅里叶序列给出：

$$\sum_{i=1}^{n} \langle \mathbf{x} \cdot \mathbf{e}_i \rangle \mathbf{e}_i$$

这自然引出在无限基情况下对序列性质的研究。

定义 B.23 如果 S 是希尔伯特空间中的一个正交集，并且没有其他的正交集可以包含 S 使其为真子集，也就是说 S 是最大的，则 S 称为 H 的一个正交基（或完备正交系统）。

定理 B.24 每个希尔伯特空间 H 都有正交基。假定 $S = \{\mathbf{x}_\alpha\}_{\alpha \in A}$ 是 H 的一个正交基。则 $\forall \mathbf{y} \in H$，有：

$$\mathbf{y} = \sum_{\alpha \in A} \langle \mathbf{y} \cdot \mathbf{x}_\alpha \rangle \mathbf{x}_\alpha$$

并且 $\|\mathbf{y}\|^2 = \sum_{\alpha \in A} |\langle \mathbf{y} \cdot \mathbf{x}_\alpha \rangle|^2$。

这个定理表明，就像在有限维空间一样，希尔伯特空间的每个元素可以表示为（可能无限的）基元素的线性组合。

系数 $\langle \mathbf{y} \cdot \mathbf{x}_\alpha \rangle$ 通常称为 $\mathbf{y}$ 对应于基 $S = \{\mathbf{x}_\alpha\}_{\alpha \in A}$ 的傅里叶系数。

例 B.25 考虑实数可数序列。希尔伯特空间ℓ_2是序列$\mathbf{z}=\{z_1,z_2,\ldots,z_i,\ldots\}$的集合，满足：

$$\|\mathbf{z}\|_2^2=\sum_{i=1}^{\infty}z_i^2<\infty$$

这里序列 $\mathbf{x}$ 和 $\mathbf{z}$ 的内积定义为：

$$\langle\mathbf{x}\cdot\mathbf{z}\rangle=\sum_{i=1}^{\infty}x_iz_i$$

如果$\boldsymbol{\mu}=\{\mu_1,\mu_2,\ldots,\mu_i,\ldots\}$是正实数可数序列，希尔伯特空间$\ell_2(\boldsymbol{\mu})$是序列$\mathbf{z}=\{z_1,z_2,\ldots,z_i,\ldots\}$的集合，满足：

$$\|\mathbf{z}\|_2^2=\sum_{i=1}^{\infty}\mu_iz_i^2<\infty$$

这里序列 $\mathbf{x}$ 和 $\mathbf{z}$ 的内积定义为：

$$\langle\mathbf{x}\cdot\mathbf{z}\rangle=\sum_{i=1}^{\infty}\mu_ix_iz_i$$

线性赋范空间ℓ_1是序列$\mathbf{z}=\{z_1,z_2,\ldots,z_i,\ldots\}$的集合，满足：

$$\|\mathbf{z}\|_1=\sum_{i=1}^{\infty}|z_i|<\infty$$

例 B.26 考虑$\mathbb{R}^n$的子集 X 上的连续实值函数集。希尔伯特空间$L_2(X)$是满足：

$$\|f\|_{L_2}=\int_X f(\mathbf{x})^2d\mathbf{x}<\infty$$

的函数 f 的集合，这里函数 f 和 g 的内积定义为：

$$\langle f\cdot g\rangle=\int_X f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$$

赋范空间$L_\infty(X)$是满足：

$$\|f\|_{L_\infty}=\sup_{\mathbf{x}\in X}|f(\mathbf{x})|<\infty$$

的函数集。

B.4 算子、特征值和特征向量

定义 B.27 从希尔伯特空间 H 到其自身的线性函数称为线性算子。对线性算子 A，

如果存在数$\|A\|$，对所有$x \in H$，满足$\|Ax\| \leqslant \|A\|\|x\|$，称$A$是有界的。

定义 B.28 令A是希尔伯特空间H上的线性算子。如果存在向量$0 \neq \mathbf{x} \in H$，对某个标量λ，满足$A\mathbf{x} = \lambda\mathbf{x}$，则称$\lambda$为$A$的特征值，相应的$\mathbf{x}$为特征向量。

定义 B.29 对在希尔伯特空间H上的有界线性算子A，如果对所有$\mathbf{x}, \mathbf{z} \in H$有：

$$\langle A\mathbf{x} \cdot \mathbf{z}\rangle = \langle \mathbf{x} \cdot A\mathbf{z}\rangle$$

称A为自伴随的。对有限维空间$\mathbb{R}^n$这意味着相应的$n \times n$矩阵$\mathbf{A}$满足$\mathbf{A} = \mathbf{A}'$，也就是$\mathbf{A}_{ij} = \mathbf{A}_{ji}$。这样的矩阵称为对称的。

下面的定理涉及称为紧的性质。这里忽略了它定义，因为尽管的确要涉及，但对于理解本书的内容不是很重要。

定理 B.30 令A为希尔伯特空间H上的自伴随紧致线性算子。则存在H的完备正交基$\{\phi_i\}_{i=1}^{\infty}$，满足：

$$A\phi_i = \lambda_i\phi_i$$

并且当$i \to \infty$有$\lambda_i \to 0$。

在有限情况下，该定理表明对称矩阵有一个正交的特征向量集。

定义 B.31 如果对称方阵的特征值都是正（非负）的，称这个矩阵为（半）正定。

命题 B.32 令$\mathbf{A}$是对称矩阵。则$\mathbf{A}$是（半）正定，当且仅当对任意向量$\mathbf{x} \neq 0$，有：

$$\mathbf{x}'\mathbf{A}\mathbf{x} > 0 \ (\geqslant 0)$$

令$\mathbf{M}$是任意矩阵（可能是非方阵），并令$\mathbf{A} = \mathbf{M}'\mathbf{M}$。则$\mathbf{A}$是一个半正定矩阵，因为对任意向量$\mathbf{x}$，可以写为：

$$\mathbf{x}'\mathbf{A}\mathbf{x} = \mathbf{x}'\mathbf{M}'\mathbf{M}\mathbf{x} = (\mathbf{M}\mathbf{x})'\mathbf{M}\mathbf{x} = \langle \mathbf{M}\mathbf{x} \cdot \mathbf{M}\mathbf{x}\rangle = \|\mathbf{M}\mathbf{x}\|^2 \geqslant 0$$

如果$\mathbf{M}$的列取为向量$\mathbf{x}_i$，$i = 1, \ldots, n$，则$\mathbf{A}$为集合$S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$的Gram矩阵，这表示Gram矩阵总是半正定的。如果S线性无关，则$\mathbf{A}$是正定的。

参 考 书 目

[1] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[2] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

[3] S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 1999. to appear.

[4] J. K. Anlauf and M. Biehl. The adatron: an adaptive perceptron algorithm. *Europhysics Letters*, 10:687–692, 1989.

[5] M. Anthony and P. Bartlett. *Learning in Neural Networks : Theoretical Foundations*. Cambridge University Press, 1999.

[6] M. Anthony and N. Biggs. *Computational Learning Theory*, volume 30 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1992.

[7] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[8] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. Hardness of approximate optima in lattices, codes and linear systems. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[9] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 43–54. MIT Press, 1999.

[10] P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

[11] M. Bazaraa, D. Sherali, and C. Shetty. *Nonlinear Programming : Theory and Algorithms.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, 1992.

[12] K. Bennett, N. Cristianini, J. Shawe-Taylor, and D. Wu. Enlarging the margin in perceptron decision trees. *Machine Learning.* to appear.

[13] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems, 12*, pages 368–374. MIT Press, 1998.

[14] K. P. Bennett and E. J. Bredensteiner. Geometry in learning. In C. Gorini, E. Hart, W. Meyer, and T. Phillips, editors, *Geometry at Work.* Mathematical Association of America, 1998.

[15] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[16] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[17] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition algorithms using realistic 3D models. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Artificial Neural Networks – ICANN'96*, pages 251 – 256. Springer Lecture Notes in Computer Science, Vol. 1112, 1996.

[18] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[20] P. S. Bradley, O. L. Mangasarian, and D. R. Musicant. Optimization methods in massive datasets. Technical Report Data Mining Institute TR-99-01, University of Wisconsin in Madison, 1999.

[21] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. Technical report, University of California in Santa Cruz, 1999. (submitted for publication).

[22] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines, 1999. [http://www.cse.ucsc.edu/research/compbio/genex/genex.html].

Santa Cruz, University of California, Department of Computer Science and Engineering.

[23] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[24] C. Campbell and N. Cristianini. Simple training algorithms for support vector machines. Technical Report CIG-TR-KA, University of Bristol, Engineering Mathematics, Computational Intelligence Group, 1999.

[25] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram-based image classification. *IEEE Transaction on Neural Networks*, 1999.

[26] C. Cortes. *Prediction of Generalization Ability in Learning Machines*. PhD thesis, Department of Computer Science, University of Rochester, USA, 1995.

[27] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[28] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14:326–334, 1965.

[29] N. Cristianini, C. Campbell, and J. Shawe-Taylor. A multiplicative updating algorithm for training support vector machine. In *Proceedings of the 6th European Symposium on Artificial Neural Networks (ESANN)*, 1999.

[30] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines: the web-site associated with the book, 2000.

[31] N. Cristianini, J. Shawe-Taylor, and C. Campbell. Dynamically adapting kernels in support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems, 11*. MIT Press, 1998.

[32] N. Cristianini, J. Shawe-Taylor, and P. Sykacek. Bayesian classifiers are large margin hyperplanes in a Hilbert space. In J. Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*, pages 109–117. Morgan Kaufmann, 1998.

[33] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of mathematics. Springer, 1996.

[34] R. Dietrich, M. Opper, and H. Sompolinsky. Statistical mechanics of support vector networks. *Physics Review Letters*, 82:2975, 1999.

[35] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[36] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *7th International Conference on Information and Knowledge Management*, 1998.

[37] T. Evgeniou and M. Pontil. On the V_γ dimension for regression in reproducing kernel Hilbert spaces. In *Algorithmic Learning Theory: ALT-99*. Springer-Verlag, 1999.

[38] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[39] T. Evgeniou, M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. Technical Report CBCL Paper #171/AI Memo #1654, Massachusetts Institute of Technology, 1999.

[40] R. Fisher. *Contributions to Mathematical Statistics*. Wiley, 1952.

[41] R. Fletcher. *Practical methods of Optimization*. Wiley, 1988.

[42] S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[43] Y. Freund and R.E. Schapire. Large margin classification using the perceptron algorithm. In J. Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*. Morgan Kaufmann, 1998.

[44] T. Friess, N. Cristianini, and C. Campbell. The kernel-Adatron: a fast and simple training procedure for support vector machines. In J. Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*. Morgan Kaufmann, 1998.

[45] T. Friess and R. Harrison. Support vector neural networks: the kernel adatron with bias and soft margin. Technical Report ACSE-TR-752, University of Sheffield, Department of ACSE, 1998.

[46] T. Friess and R. Harrison. A kernel based adaline. In *Proceedings of the 6th European Symposium on Artificial Neural Networks (ESANN)*, 1999.

[47] S. I. Gallant. Perceptron based learning algorithms. *IEEE Transactions on Neural Networks*, 1:179–191, 1990.

[48] C. Gentile and M. K. Warmuth. Linear hinge loss and average margin. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[49] M. Gibbs and D. MacKay. Efficient implementation of Gaussian processes. Technical report, Department of Physics, Cavendish Laboratory, Cambridge University, UK, 1997.

[50] M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification.* PhD thesis, University of Cambridge, 1997.

[51] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

[52] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[53] GMD-FIRST. GMD-FIRST web site on Support Vector Machines. http://svm.first.gmd.de.

[54] L. Gurvits. A note on a scale-sensitive dimension of linear bounded functionals in Banach spaces. In *Proceedings of Algorithmic Learning Theory, ALT-97*, 1997.

[55] I. Guyon, N. Matić, and V. Vapnik. Discovering informative patterns and data cleaning. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smythand, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 181–203. MIT Press, 1996.

[56] I. Guyon and D. Stork. Linear discriminant and support vector classifiers. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers.* MIT Press, 1999.

[57] M. H. Hassoun. *Optical Threshold Gates and Logical Signal Processing.* PhD thesis, Department of ECE, Wayne State University, Detroit, USA, 1986.

[58] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California in Santa Cruz, Computer Science Department, July 1999.

[59] R. Herbrich, T. Graepel, and C. Campbell. Bayes point machines: Estimating the Bayes point in kernel space. In *Proceedings of IJCAI Workshop Support Vector Machines*, 1999.

[60] R. Herbrich, K. Obermayer, and T. Graepel. Large margin rank boundaries for ordinal regression. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers.* MIT Press, 1999.

[61] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[62] C. Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4:79–85, 1957.

[63] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[64] K. U. Höffgen, K. S. van Horn, and H. U. Simon. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50(1):114-125, 1995.

[65] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems, 11*. MIT Press, 1998.

[66] T. S. Jaakkola and D. Haussler. Probabilistic kernel regression models. In *Proceedings of the 1999 Conference on AI and Statistics*, 1999.

[67] T. Joachims. Text categorization with support vector machines. In *Proceedings of European Conference on Machine Learning (ECML)*, 1998.

[68] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–184. MIT Press, 1999.

[69] W. Karush. *Minima of Functions of Several Variables with Inequalities as Side Constraints*. Department of Mathematics, University of Chicago, 1939. MSc Thesis.

[70] L. Kaufmann. Solving the quadratic programming problem arising in support vector classification. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 147–168. MIT Press, 1999.

[71] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[72] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Science*, 48(3):464–497, 1994. Earlier version appeared in FOCS90.

[73] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. Technical report, Department of CSA, IISc, Bangalore, India, 1999. Technical Report No. TR-ISL-99-03.

[74] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. Technical report, Control Division, Department of Mechanical and Production Engineering, National University of Singapore, 1999. Technical Report No. CD-99-14.

[75] J. Kivinen and M. K. Warmuth. Additive versus exponentiated gradient for linear prediction. *Information and Computation*, 132:1–64, 1997.

[76] A. Kowalczyk. Maximal margin perceptron. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[77] H. Kuhn and A. Tucker. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, pages 481–492. University of California Press, 1951.

[78] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman-Soulié and P. Gallinari, editors, *Proceedings ICANN'95 – International Conference on Artificial Neural Networks*, volume II, pages 53–60. EC2, 1995.

[79] N. Littlestone and M. Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.

[80] D. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[81] D. MacKay. Introduction to Gaussian processes. In *Neural Networks and Machine Learning (NATO Asi Series); Ed. by Chris Bishop*, 1999.

[82] D. J. C. MacKay. A practical Bayesian framework for backprop networks. *Neural Computation*, 4:448–472, 1992.

[83] O. Mangasarian. Generalized support vector machines. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[84] O. L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

[85] O. L. Mangasarian. Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, IT-14:801–807, 1968.

[86] O. L. Mangasarian. *Nonlinear Programming*. SIAM, 1994.

[87] O. L. Mangasarian. Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 42(1):183–201, 1997.

[88] O. L. Mangasarian. Generalized support vector machines. Technical Report Mathematical Programming TR 98-14, University of Wisconsin in Madison, 1998.

[89] O. L. Mangasarian and D. R. Musicant. Successive overrelaxation for support vector machines. Technical Report Mathematical Programming TR 98-18, University of Wisconsin in Madison, 1998.

[90] O. L. Mangasarian and D. R. Musicant. Data discrimination via nonlinear generalized support vector machines. Technical Report Mathematical Programming TR 99-03, University of Wisconsin in Madison, 1999.

[91] O. L. Mangasarian and D. R. Musicant. Massive support vector regression. Technical Report Data Mining Institute TR-99-02, University of Wisconsin in Madison, 1999.

[92] MATLAB. *User's Guide*. The MathWorks, Inc., 1992.

[93] David A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 230–234. ACM Press, 1998.

[94] David A. McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 164–170. ACM Press, 1999.

[95] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A 209:415–446, 1909.

[96] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[97] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[98] M.L. Minsky and S.A. Papert. *Perceptrons*. MIT Press, 1969. Expanded Edition 1990.

[99] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[100] J. J. More and S. J. Wright. *Optimization Software Guide*. Frontiers in Applied Mathematics, Volume 14. Society for Industrial and Applied Mathematics (SIAM), 1993.

[101] B. A. Murtagh and M. A. Saunders. MINOS 5.4 user's guide. Technical Report SOL 83.20, Stanford University, 1993.

[102] R. Neal. *Bayesian Learning in Neural Networks*. Springer Verlag, 1996.

[103] R. M. Neal. Monte carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report TR 9702, Department of Statistics, University of Toronto, 1997.

[104] A. B. Novikoff. On convergence proofs on perceptrons. In *Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622. Polytechnic Institute of Brooklyn, 1962.

[105] M. Opper and W. Kinzel. Physics of generalization. In E. Domany J. L. van Hemmen and K. Schulten, editors, *Physics of Neural Networks III*. Springer Verlag, 1996.

[106] M. Opper and F. Vivarelli. General bounds on Bayes errors for regression with Gaussian processes. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems, 11*. MIT Press, 1998.

[107] M. Opper and O. Winther. Gaussian processes and SVM: Mean field and leave-one-out. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[108] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[109] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276–285. IEEE, 1997.

[110] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proceedings of Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[111] E. Osuna and F. Girosi. Reducing run-time complexity in SVMs. In *Proceedings of the 14th International Conference on Pattern Recognition, Brisbane, Australia*, 1998. To appear.

[112] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, 1999.

[113] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In *Neural Information Processing Systems (NIPS 99)*, 1999. to appear.

[114] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[115] T. Poggio. On optimal nonlinear associative recall. *Biological Cybernetics*, 19:201–209, 1975.

[116] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), September 1990.

[117] D. Pollard. *Convergence of Stochastic Processes*. Springer, 1984.

[118] M. Pontil and A. Verri. Object recognition with support vector machines. *IEEE Trans. on PAMI*, 20:637–646, 1998.

[119] K. Popper. *The Logic of Scientific Discovery*. Springer, 1934. First English Edition by Hutchinson, 1959.

[120] C. Rasmussen. *Evaluation of Gaussian Processes and Other Methods for Non-Linear Regression*. PhD thesis, Department of Computer Science, University of Toronto, 1996. ftp://ftp.cs.toronto.edu/pub/carl/thesis.ps.gz.

[121] R. Rifkin, M. Pontil, and A. Verri. A note on support vector machines degeneracy. Department of Mathematical Sciences CBCL Paper #177/AI Memo #1661, Massachusetts Institute of Technology, June 1999.

[122] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1959.

[123] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, 1988.

[124] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development*, 49:210–229, 1959.

[125] C. Saunders, A. Gammermann, and V. Vovk. Ridge regression learning algorithm in dual variables. In J. Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*. Morgan Kaufmann, 1998.

[126] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola. Support vector machine - reference manual. Technical Report CSD-TR-98-03, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998. TR available as http://www.dcs.rhbnc.ac.uk/research/compint/areas/comp_learn/sv/pub/report98-03.ps; SVM available at http://svm.dcs.rhbnc.ac.uk/.

[127] R. Schapire, Y. Freund, P. Bartlett, and W. Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

[128] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining*. AAAI Press, 1995.

[129] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, 1997.

[130] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Shrinking the tube: a new support vector regression algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems, 11*. MIT Press, 1998.

[131] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Support vector regression with automatic accuracy control. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, pages 147 – 152. Springer Verlag, 1998.

[132] B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1999.

[133] B. Schölkopf, J. Shawe-Taylor, A. Smola, and R. Williamson. Generalization bounds via the eigenvalues of the gram matrix. Technical Report NC-TR-1999-035, NeuroCOLT Working Group, http://www.neurocolt.com, 1999.

[134] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks – ICANN'97*, pages 583–588. Springer Lecture Notes in Computer Science, Volume 1327, 1997.

[135] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. Technical Report NC-TR-98-031, NeuroCOLT Working Group, http://www.neurocolt.com, 1998.

[136] B. Schölkopf, A. J. Smola, and K. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 327–352. MIT Press, 1999.

[137] J. Shawe-Taylor. Classification accuracy based on observed margin. *Algorithmica*, 22:157–172, 1998.

[138] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

[139] J. Shawe-Taylor and N. Cristianini. Robust bounds on generalization from the margin distribution. Technical Report NC-TR-1998-020, NeuroCOLT Working Group, http://www.neurocolt.com, 1998.

[140] J. Shawe-Taylor and N. Cristianini. Further results on the margin distribution. In *Proceedings of the Conference on Computational Learning Theory, COLT 99*, pages 278–285, 1999.

[141] J. Shawe-Taylor and N. Cristianini. Margin distribution and soft margin. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[142] J. Shawe-Taylor and N. Cristianini. Margin distribution bounds on generalization. In *Proceedings of the European Conference on Computational Learning Theory, EuroCOLT'99*, pages 263–273, 1999.

[143] F. W. Smith. Pattern classifier design by linear programming. *IEEE Transactions on Computers*, C-17:367–372, 1968.

[144] A. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22:211–231, 1998.

[145] A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 1998. Invited paper, in press.

[146] A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

[147] A. Smola, B. Schölkopf, and K.-R. Müller. General cost functions for support vector regression. In T. Downs, M. Frean, and M. Gallagher, editors, *Proc. of the Ninth Australian Conf. on Neural Networks*, pages 79–83. University of Queensland, 1998.

[148] A. J. Smola. *Learning with Kernels*. PhD thesis, Technische Universität Berlin, 1998.

[149] A. J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans. *Advances in Large Margin Classifiers*. MIT Press, 1999.

[150] P. Sollich. Learning curves for Gaussian processes. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems, 11*. MIT Press, 1998.

[151] R. J. Solomonoff. A formal theory of inductive inference: Part 1. *Inform. Control*, 7:1–22, 1964.

[152] R. J. Solomonoff. A formal theory of inductive inference: Part 2. *Inform. Control*, 7:224–254, 1964.

[153] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, 1977.

[154] A. M. Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

[155] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, Nov 1984.

[156] R. J. Vanderbei. LOQO user's manual – version 3.10. Technical Report SOR-97-08, Princeton University, Statistics and Operations Research, 1997. Code available at http://www.princeton.edu/ rvdb/.

[157] V. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, 1979. (English translation Springer Verlag, 1982).

[158] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[159] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[160] V. Vapnik and O. Chapelle. Bounds on error expectation for SVM. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[161] V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

[162] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[163] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie-Verlag, Berlin, 1979).

[164] V. Vapnik and A. Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26(3):532–553, 1981.

[165] V. Vapnik and A. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

[166] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.

[167] V. Vapnik and S. Mukherjee. Support vector method for multivariant density estimation. In *Neural Information Processing Systems (NIPS 99)*, 1999. to appear.

[168] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of IJCAI Workshop Support Vector Machines*, 1999.

[169] M. Vidyasagar. *A Theory of Learning and Generalization*. Springer, 1997.

[170] S. Vijayakumar and S. Wu. Sequential support vector classifiers and regression. In *Proceedings of the International Conference on Soft Computing (SOCO'99)*, pages 610–619, 1999.

[171] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1990.

[172] G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 69–88. MIT Press, 1999.

[173] G. Wahba, Y. Lin, and H. Zhang. GACV for support vector machines. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[174] C. Watkins. Dynamic alignment kernels. Technical Report CSD-TR-98-11, Royal Holloway, University of London, Computer Science department, January 1999.

[175] C. Watkins. Dynamic alignment kernels. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[176] C. Watkins. Kernels from matching operations. Technical Report CSD-TR-98-07, Royal Holloway, University of London, Computer Science Department, July 1999.

[177] J. Weston and R. Herbrich. Adaptive margin support vector machines. In A.J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[178] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of the 6th European Symposium on Artificial Neural Networks (ESANN)*, 1999.

[179] B. Widrow and M. Hoff. Adaptive switching circuits. *IRE WESCON Convention record*, 4:96–104, 1960.

[180] C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*. Kluwer, 1998.